

Trajektorien- und Aufgabenplanung kooperierender Fahrzeuge: Diskret-kontinuierliche Modellierung und Optimierung

Vom Fachbereich Informatik der
Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)

von

Dipl.-Math. Christian Reinl
(geboren in Weiden in der Oberpfalz)

Referenten der Arbeit: Prof. Dr. rer. nat. Oskar von Stryk
Prof. Dr.-Ing. Olaf Stursberg
(Universität Kassel)

Tag der Einreichung: 09.02.2010
Tag der mündlichen Prüfung: 25.03.2010

Darmstadt 2010

D 17

Vorwort

Die vorliegende Arbeit wurde im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter im Forschungszentrum *Computational Engineering* und als Stipendiat im DFG-Graduiertenkolleg *Cooperative, Adaptive and Responsive Monitoring in Mixed Mode Environments* am Fachbereich Informatik der Technischen Universität Darmstadt angefertigt.

Am Fachgebiet *Simulation, Systemoptimierung und Robotik* durfte ich dabei in besonderem Maß von einer erfolgsorientierten Atmosphäre, förderlichen Rahmenbedingungen, dem breiten Wissensspektrum von Kollegen und der Mitwirkung an verschiedensten Aktivitäten modernen wissenschaftlichen Arbeitens profitieren.

Die Mitgliedschaft im Graduiertenkolleg garantierte mir dabei einerseits die gesicherte Finanzierung des Promotionsvorhaben und bot mir ein Umfeld des fruchtbaren interdisziplinären Austauschs mit Ingenieuren, Informatikern und Mathematikern unterschiedlicher Spezialisierung. Sowohl auf diese Erfahrungen als auch auf Gelegenheiten außerfachliche Qualifikationen im Rahmen von Organisationstätigkeiten, Projektarbeit und Kursangeboten zu trainieren, blicke ich dankbar zurück.

Im Rückblick auf prägende Jahre der Promotion möchte ich an dieser Stelle meinen Dank aussprechen an alle, die mich begleitet, unterstützt und gefördert haben. Dazu zählen Studierende, Kollegen und Bekannte mit denen ich in Diskussionen Zuversicht, Einsicht und Klarheit finden durfte. Insbesondere denke ich dabei an Kollegen, mit denen ich in den letzten Jahren das Büro teilen durfte: Markus Glocker mit großem Erfahrungsschatz diskret-kontinuierlicher Optimalsteuerung; Thomas Hemker als erfahrener Experte für Optimierungsmethoden und geschätztes Vorbild strukturierten Arbeitens; Max Risler mit reicher Anwendungserfahrung für Mehrrobotersysteme und Juliane Kuhn, welche stets eine interessierte und engagierte Diskussionspartnerin war und nun die Thematik kooperativer Fahrzeugsysteme für ihr eigenes Promotionsvorhaben aufgegriffen hat.

Besonderer Dank gilt dem Referenten der Arbeit und Leiter der Arbeitsgruppe, Professor Dr. Oskar von Stryk, der sich stets um gute Rahmenbedingungen bemühte, viele Freiheiten gewährte und mit Erfahrung und interdisziplinärem Weitblick mich auf diesem Weg betreute. Auch Professor Dr.-Ing. Olaf Stursberg möchte ich für wertvolle Anregungen und die Übernahme des Korreferats danken.

In meiner Rückschau auf diesen Weg möchte besonders auch denen „Danke!“ sagen, die mich in ganz unterschiedlicher Weise begleitet haben, mir Stütze und Orientierung waren. Insbesondere denke ich dabei an meine Geschwister, meine Eltern und Schwiegereltern, meine Frau Katharina und alle Freunde, die mir mit Verständnis und Rücksicht begegnet sind, den Rücken freigehalten haben und mir dabei geholfen haben, den Kontakt zu den wirklich wichtigen Dingen im Leben nicht zu verlieren. Vielen Dank!

Inhaltsverzeichnis

Kurzdarstellung	VII
1 Problemstellung und Ziele der Arbeit	1
1.1 Problemcharakteristik und Vorgehensweise	1
1.2 Einordnung in Methoden der Aufgabenzuweisung und Bahnplanung	3
1.3 Ziele und wissenschaftliche Beiträge dieser Arbeit	6
2 Beschreibung und Analyse hybrider dynamischer Systeme	8
2.1 Grundlegende Merkmale hybrider dynamischer Systeme	9
2.1.1 Logische Bedingungen	10
2.1.2 Gemischt-logische Bedingungen	12
2.2 Modellierung hybrider dynamischer Systeme	14
2.2.1 Hybride Zustandsautomaten	14
2.2.2 Vorschlag zur Erweiterung auf Modelle mit hierarchischer Struktur	15
2.2.3 Anwendung auf Mehrfahrzeugsysteme	16
2.3 Zeitdiskrete lineare hybride Systeme	18
2.3.1 Gemischt logisch-dynamische Beschreibung (MLD)	19
2.3.2 Äquivalente Alternativen zu MLD-Beschreibungen	21
2.4 Kenngrößen und Methoden der Analyse hybrider Systeme	22
2.4.1 Methoden der Informatik	22
2.4.2 Methoden der Regelungstheorie	23
2.4.3 Integrationsvorschläge zu Ansätzen der Informatik und mathemati- scher Optimierung	25
3 Optimalsteuerung diskret-kontinuierlicher Systeme	27
3.1 Nichtlineare Systeme	27
3.1.1 Eigenschaften und Merkmale	27
3.1.2 Transformation eines Automatenmodells	31
3.1.3 Numerische Verfahren	32
3.2 Konzepte zur Modellapproximation	34
3.2.1 Hybridisierungskonzepte nichtlinearer Systemanteile	35
3.2.2 Zeitdiskretisierung und Euler-Approximation	37
3.2.3 Vorschläge zum Einsatz von approximativen Modellen	38
3.3 Modellprädiktive Regelung (MPC) diskret-kontinuierlicher linearer Modelle	40
3.3.1 Prinzip der modellprädiktiven Regelung	40
3.3.2 Gemischt-ganzzahlige lineare Optimierung (MILP)	42
3.3.3 Algorithmen und Eigenschaften MILP-basierter MPC-Methoden . .	43
3.4 Weiterführende Konzepte	47
3.4.1 Iterative Verbesserung der Optimallösung	47

3.4.2	Anwendung gemischt-ganzzahliger MPC in kooperativen Fahrzeugsystemen	48
4	Modellierung kooperativer Mehrfahrzeugsysteme	51
4.1	Allgemeines Modellierungskonzept	51
4.2	Modellierung und Bewertung kooperativer Mobilität	52
4.2.1	Fahrzeugdynamik	52
4.2.2	Zeitdiskrete, lineare Formulierung einfacher Fahrdynamiken	55
4.2.3	Bewertung kooperativer Bewegung durch eine Zielfunktion	59
4.3	Paarweise Beziehungen und Formationsstrukturen	60
4.3.1	Paarweise Beziehungen	60
4.3.2	Veränderliche Graphenstrukturen:	63
4.3.3	Spezialfall: Mobilität im dichten drahtlosen Sensornetzwerk	65
4.4	Möglichkeiten der Auskopplung von Teilproblemen	69
4.4.1	Grundproblem Aufgabenzuweisung	70
4.4.2	Handlungsreisendenprobleme	70
5	Anwendungen und Ergebnisse	72
5.1	Benchmarkproblem der Zielzuweisung zur kooperativen Erkundung	74
5.1.1	Untersuchung des Grundproblems	75
5.1.2	Untersuchung von Detailspekten der Modellierung und Approximation	80
5.2	Mobilität, Kommunikation und Konnektivität	84
5.2.1	Kooperative Mobilität im Drahtlossensornetzwerk	84
5.2.2	Kooperative Mehrfahrzeugsysteme mit Kommunikationsbeschränkungen	87
5.3	Kooperatives Verhalten in dynamischer Umgebung	92
5.3.1	Taktik und Trajektorienplanung im Roboterfußball	93
5.3.2	Kooperative Zielbeobachtung	99
5.4	Aspekte der Übertragung auf reale Mehrfahrzeugsysteme	103
6	Zusammenfassung und Ausblick	105
Anhang		107
A	Zentrale Sätze der Optimalsteuerung	107
B	Definitionen zur Systemstabilität (nach [Laz06])	108
Literaturverzeichnis		110

Kurzdarstellung

Im Kern der Aufgabenzuweisung und Trajektorienplanung für kooperierende Fahrzeuge stehen schwer entscheidbare Fragestellungen. Verschiedenartige Aufgaben, deren Erfüllung maßgeblich von physikalischer Bewegungsdynamik abhängt, führen in der Systemanalyse und in der Entscheidungsfindung zu einer engen Kopplung diskreter Struktur und kontinuierlicher Dynamik.

In praktischen Anwendungen werden bislang Heuristiken für sehr spezifische Problemstellungen eingesetzt, oder man berücksichtigt die nichtlinearen bewegungsdynamischen Eigenschaften nur sehr grob in den Planungsmethoden. Eine Verallgemeinerung und Übertragung bestehender Ansätze auf andere Fahrzeugklassen und Aufgabenstellungen ist deshalb meist nur eingeschränkt realisierbar.

In einem beständig wachsenden Feld neuer Anwendungen kooperativer, autonomer Mehrfahrzeugsysteme kommt der Entwicklung einer möglichst breit anwendbaren Methodik eine wichtige Bedeutung zu. Ein entsprechendes Konzept muss die wesentlichen Systemmerkmale in der Modellierung und Optimalplanung zur Anwendung in der Systemauslegung und der Reglerentwicklung berücksichtigen.

In der vorliegenden Arbeit wird dazu ein in sich konsistentes Modellierungs-, Approximations- und Optimierungskonzept vorgestellt, das auf der Theorie hybrider dynamischer Systeme, der mathematischen nichtlinearen gemischt-ganzzahligen Optimalsteuerung und auf modellprädiktive Methoden der Regelungstheorie aufbaut.

Mit Hilfe hierarchischer hybrider Zustandsautomaten wird die enge Kopplung diskreter und kontinuierlicher Systemdynamik im Modell erfasst und über geeignete Transformationen der mathematischen Optimierung zugänglich gemacht. Dabei werden insbesondere lineare Approximationen betrachtet, die das Potential günstiger Rechenzeiten und globaler Optimalität für die Ersatzmodelle besitzen.

Die Lösung der entstehenden diskret-kontinuierlichen Optimierungsprobleme erlaubt für viele Fragestellungen kooperativen Verhaltens eine effizient berechenbare Näherungslösung und kann in der vorliegenden Form in Spezialfällen bereits zu einer echtzeitfähigen Regelung der Aufgabenzuweisung verwendet werden.

Für repräsentative Benchmarkszenarien und neuartige Fragestellungen – wie zur Aufrechterhaltung drahtloser Kommunikation zwischen Fahrzeugen – werden numerische Ergebnisse präsentiert, welche die Leistungsfähigkeit der Konzepte demonstrieren und deren Grenzen ausloten. Mit Hilfe der vorliegenden Arbeit ist es möglich, Abschätzungen zur Systemauslegung und zur Entwicklung heuristischer Reglerkonzepte für das Kernproblem kooperativer Mobilität zu berechnen, unter Berücksichtigung physikalischer Bewegungsdynamik und der charakteristischen diskret-kontinuierlichen Kopplung von Systemzuständen.

1 Problemstellung und Ziele der Arbeit

Mobile autonome Systeme nehmen seit einigen Jahren einen breiten – nach wie vor wachsenden – Bereich in Forschung und Entwicklung ein. Dabei reicht die Entwicklung der Systeme bereits hin zu weitestgehend autonom agierenden Servicerobotern in Such- und Rettungseinsätzen, in industrieller Fertigung und in privaten Haushalten.

Die Kooperation mehrerer mobiler autonomer Systeme stellt die Forschung dabei vor besondere Herausforderungen, birgt allerdings auch ein großes Potential an Synergien in sich. So kann die erhöhte Zahl teilnehmender Robotersysteme (z.B. in Katastropheneinsätzen) nicht nur die Zeit zur Aufgabenerfüllung verringern, sondern kann darüber hinaus für eine höhere Ausfallsicherheit des Gesamtsystems sorgen oder die Erfüllung bestimmter Teilaufgaben unter Umständen überhaupt erst möglich machen.

Die vorliegende Arbeit widmet sich dabei der Kernproblematik mobiler, kooperativer autonomer Systeme, in denen die Mobilität der beteiligten Roboter oder unbemannten Fahrzeuge die Erfüllung der gestellten Systemaufgaben entscheidend beeinflusst. Das *Grundproblem kooperativer Mobilität* ist durch eine enge Kopplung diskreter Entscheidung (z. B. der Aufgabenzuweisung) und der Planung möglichst optimaler, kontinuierlicher Bewegungstrajektorien charakterisiert.

Die bestehenden Ansätze und Methoden zur Steuerung von kooperativen Mehrfahrzeug- und Mehrroboter-Systemen sind äußerst vielfältig und hängen vorwiegend von dem Zusammenwirken bestimmter Fahrzeugklassen zur Erfüllung konkreter Aufgaben und von einer unter Umständen sehr spezifischen Verfügbarkeit von Sensordaten ab.

Die Forschung und Entwicklung ist hier an einer Problembetrachtung auf verschiedenen Ebenen interessiert. So kann beispielsweise eine detaillierte physikalische Modellierung des Gesamtsystems für wichtige Abschätzungen in der Systemauslegung verwendet werden, während sehr abstrahierte und idealisierte Approximationen einer Teilfunktionalität zur Entwicklung einer dezentral organisierten Entscheidungsarchitektur herangezogen werden.

Die Motivation dieser Arbeit besteht einerseits in der Erhöhung der Autonomie in kooperativen Fahrzeugsystemen durch eine allgemeine, effiziente Methode der simultanen Aufgabenverteilung und Trajektorienplanung und andererseits in der Bereitstellung eines möglichst allgemein anwendbaren Approximations- und Optimierungskonzeptes zur Analyse und Auslegung kooperativer Fahrzeugsysteme.

1.1 Problemcharakteristik und Vorgehensweise

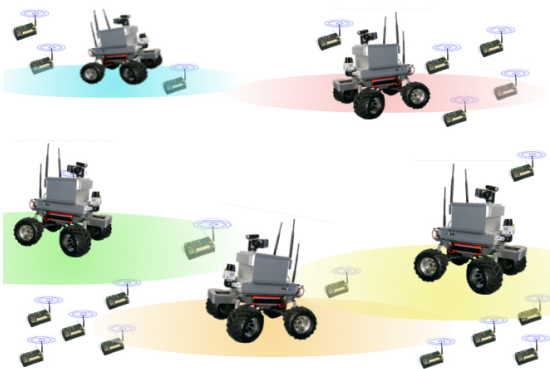
Es werden Mehrfahrzeugsysteme und Fragestellungen betrachtet, bei denen die Bestimmung eines optimalen kooperativen Verhaltens die Berücksichtigung einer engen Kopplung von diskreten Ereignissen mit kontinuierlichen Fahrdynamiken erfordert. In Abb. 1.1 sind einige Beispiele dargestellt, welche diese Problemcharakteristik aufweisen.



(a) Erkundung von Waldbrandgebieten



(b) Kooperierende mobile Roboter im Fußball-Wettkampf



(c) Kooperierende mobile Plattformen zur Unterstützung von Drahtlos-Sensornetzwerken



(d) Automatisierter Waretransport

Abbildung 1.1: Einsatz kooperativer Systeme in Forschung und Industrie

Die zu untersuchende Problemstellung übersteigt damit deutlich die Komplexität einfacher Koordinations- und einfacher Bahnplanungsaufgaben, derer man sich häufig – bei geeigneten Voraussetzungen – im Rahmen heuristischer Lösungsansätze bedient.

Exemplarisch werden zur Motivation zunächst zwei repräsentative Szenarien mit typischen Fragestellungen betrachtet.

Szenario 1: Luftfahrzeuge zur kooperativen Erkundung (Abb. 1.1a). Eine Standardaufgabe für eine Gruppe unbemannter Luftfahrzeuge (unmanned aerial vehicles, UAVs), etwa in der Waldbrandüberwachung, Verkehrsbeobachtung oder der Aufklärung von Schadstoffausbreitungen, ist es, eine bestimmte Zahl an Wegpunkten oder Zielen und zugehörigen Flugtrajektorien so zuzuweisen, dass eine auf das Gesamtsystem bezogene Zielfunktion optimiert wird.

Abhängig vom Szenario kann Optimalität beispielsweise bezüglich der Gesamtzeit einer Aufgabenerfüllung, der dafür benötigten Energie oder über das Eintreten bestimmter Zustandswechsel gefordert werden. Häufig beschreibt eine Kombination dieser und weiterer spezifischer Größen das gewünschte Systemverhalten.

An Mehrfahrzeugsysteme werden darüber hinaus oftmals noch weitere Bedingungen gestellt, wie etwa eine diskrete Graphenstruktur, die bestimmte Formationen oder Konnektivitätsanforderungen einer Drahtloskommunikation beschreiben. Die optimale Wahl der

Abfolge von Netzwerktopologien oder Formationen und die Berechnung zugehöriger optimaler Steuersequenzen der Flugtrajektorien fallen in den Bereich der vorgestellten Arbeit.

Eine typische Fragestellung ist etwa: „*Wieviel Zeit benötigt eine Flotte von Luftfahrzeugen mindestens zur kooperativen Erkundung eines gegebenen Gebietes?*“ oder „*Wie lässt sich auf Basis der Fahrzeugdynamikeigenschaften eine optimale Zuordnung und Reihenfolge von Beobachtungsgebieten finden?*“

Szenario 2: Kooperativer Spielzug im Roboterfußball (Abb. 1.1b). Ein wissenschaftlich anerkanntes Benchmarkproblem der Kooperation mobiler, autonomer Systeme ist Roboterfußball. Insbesondere stellt es ein Testfeld der künstlichen Intelligenz für mobile, autonome Agenten in dynamischen Umgebungen dar.

Fragen nach der optimalen Taktik können hier nicht ohne weiteres aus menschlicher Erfahrung abgeleitet werden. Vielmehr können diese Fragen nur unter Berücksichtigung spezifischer Roboter-Eigenschaften wie der Bewegungsdynamik und unter konkreten Rollen- und Aufgabenbeschränkungen beantwortet werden.

Eine typische Fragestellung ist etwa „*Wie kann ein Team kooperierender mobiler Fußball-Roboter optimale Spielzüge planen?*“ oder „*Wie kann der Ball in einer vorgegebenen Zeitspanne dem gegenüberliegenden Tor möglichst nahe gebracht werden?*“.

Methodisch wird in dieser Arbeit der Einsatz möglichst global optimierender Verfahren angestrebt, so dass sich die Lösung einer konkreten Analyse- oder Steuerungs-Fragestellung schließlich stets durch eine Optimalitätseigenschaft auszeichnet und sie damit gegenüber heuristisch gewonnenen Lösungen überlegen sein lässt.

In den zu untersuchenden Fragestellungen wird stets das Gesamtsystem mit zentraler Informationsstruktur betrachtet. Die Methodik der Arbeit folgt einem systematischen Entwicklungsprozess (vgl. Abb. 1.2). Ausgehend von einer fundierten Betrachtung der Theorie hybrider dynamischer Systeme (Kapitel 2) und der diskret-kontinuierlichen Optimalsteuerung (Kapitel 3) wird auf Basis eines Modellierungs- und Approximationsprozesses eine Systembeschreibung aufgebaut (Kapitel 4), welche der Optimierung zugänglich ist.

Damit lassen sich einerseits neue Fragen der Systemanalyse beantworten und andererseits echtzeitfähige Einbettungen in modellprädiktive Reglerstrukturen realer Systemen (Kapitel 5) erreichen.

1.2 Einordnung in Methoden der Aufgabenzuweisung und Bahnplanung

Zur Aufgabenzuweisung und Bahnplanung kooperativer Mehrfahrzeug- und Mehrroboter-Systeme unterscheidet man grundsätzlich zentral und dezentral organisierte Methoden. In der zentralen Planung gibt es eine ausgezeichnete Systemkomponente, welche für die Entscheidung und Zuweisung zuständig ist, während in dezentralen Ansätzen die Entscheidungen in den einzelnen Robotern und Fahrzeugen getroffen werden. Einen guten allgemeinen Überblick über bestehende Ansätze der letzten Jahre bietet der Artikel [Mur07].

Zentrale Ansätze (Bsp: [RH02, Ear05]) haben den Vorteil, dass sie alle zur Entscheidung nötigen Informationen in einem Modell vereinen und somit die Möglichkeit haben, ein global optimales kooperatives Verhalten zu berechnen. Kooperative Aufgaben können

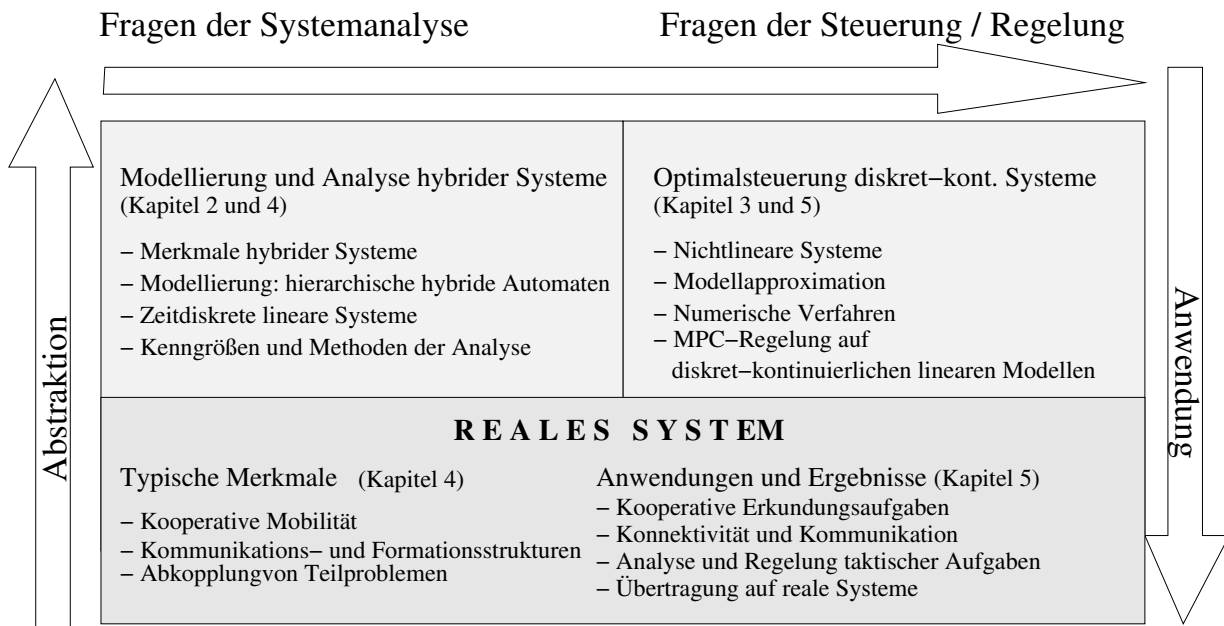


Abbildung 1.2: Methodik der Arbeit: Zunächst wird mit Hilfe abstrahierender Beschreibung auf Basis der physikalischen Systemdynamik modelliert, darauf aufbauend geeignet approximiert und somit das Problem effizienten Optimierungsmethoden zugänglich gemacht. Mit fortschreitendem Entwicklungsprozess werden zunehmend Fragen der Steuerung betrachtet mit dem Ziel, diese zur Regelung der Kooperation realer Systeme einzusetzen.

leicht beschrieben und Zuweisungen berechnet werden, welche schließlich zur Umsetzung an die beteiligten Fahrzeuge übermittelt werden. Die zentrale Entscheidungsarchitektur erlaubt auf einfache Weise die Einbindung eines menschlichen Planers, um gegebenenfalls das System in der Aufgabenzuweisung und -durchführung zu beeinflussen.

Der Nachteil einer zentralen Entscheidungsarchitektur ist die hohe Abhängigkeit von einer einzelnen Steuerungsinstanz und der zugehörigen Kommunikationsinfrastruktur. Fällt die zentrale Entscheidungskomponente aus oder ist die Kommunikation gestört, ist das gesamte System nicht weiter handlungsfähig.

In *dezentralen Architekturen* werden Entscheidungen lokal auf Basis lokaler Informationen getroffen. Damit ist die einzelne Systemkomponente weitestgehend autonom und plant eigenständig Aktionen. Dieser Aspekt erlaubt es auch, ein System wesentlich einfacher um neue Fahrzeuge zu erweitern oder Ausfälle zu kompensieren. Zur Durchführung kooperativer Aufgaben ist jedoch nach wie vor ein gewisser Informationsaustausch notwendig, wofür zahlreiche Methoden vorgestellt wurden. Dazu zählen insbesondere marktbasierende Ansätze [DS01], Publish/Subscribe-Architekturen [GM02] und komplett verhaltensbasierte Softwarearchitekturen [Par98]. Ein Rahmenkonzept, das hybride Systeme zur Rollenzuweisung für kooperative Mehrrobotersysteme betrachtet, wurde in [CKC04] vorgestellt. [Par08] bietet dazu einen umfassenden aktuellen Überblick.

Eng verwandt mit vollständig dezentralen Strukturen ist das Konzept der *verteilten Steuerung* (siehe z.B. [BCM09]). Hier werden Teilprobleme derart lokal gelöst und Informationen mit den übrigen Teilnehmern ausgetauscht, dass mittelfristig das Gesamtproblem optimal gelöst wird. Auch hier besteht der Hauptvorteil darin, dass der Ausfall einzelner Teilsysteme die Kooperationsaufgabe nicht zwangsläufig gefährdet.

Das Spektrum bestehender Entscheidungsarchitekturen verläuft kontinuierlich zwischen zentralen, dezentralen oder verteilten Ansätzen, da in Abhängigkeit von Aufgaben, der Umgebung und den eingesetzten Komponenten sich oft eine wohlüberlegte Differenzierung zwischen globaler und lokaler Planung als zielführend erweist (Bsp. [FTEBJ06]). Insbesondere eine hierarchische Gliederung der Planung und Regelung kann dabei dem Problemcharakter gut entsprechen (Bsp. [FBDWD04, GHAL05, BJSS07, DS09a]).

In der Literatur vorgestellte Ansätze zur Planung kooperativen Bewegungsverhaltens sind überwiegend entweder nur zur Lösung sehr spezieller Probleme geeignet (z.B. [RH02, BIP05, Ear05, DFL⁺08]) oder berücksichtigen lediglich heuristische Entscheidungskriterien auf Basis einer äußerst groben Approximation physikalischer Bewegungsdynamik (Bsp.[Luc02, Ris09]).

Das Problem der Zuweisung kooperativer Aufgaben, welche vorwiegend durch die Bewegungsdynamik von Fahrzeugen charakterisiert sind, zeichnet sich durch einen eng gekoppelten kombinatorischen und dynamischen Anteil aus. Systemtheoretisch sind die Probleme deshalb der Klasse *hybrider dynamischer Systeme* zuzuordnen.

Der kombinatorische Problemanteil wurde beispielsweise mit genetischen Algorithmen [SRS05] gelöst oder mit konventionellen multi-TSP-Problemformulierungen als Teil einer mehrschichtigen Entscheidungsarchitektur für eine heterogene Gruppe von Luftfahrzeugen [GHAL05] untersucht. Methoden zur Lösung allgemeiner nichtlinearer diskret-kontinuierlicher Optimalsteuerungsprobleme wurden z.B. in [vSG00, Glo05, RD05] vorgestellt.

Demgegenüber bedient sich eine Vielzahl weiterer Konzepte der gemischt-ganzzahligen linearen Optimierung (mixed-integer linear programming, MILP). Dank der Verfügbarkeit sehr effizienter MILP-Solver können selbst für relativ große Probleme in vertretbaren Rechenzeiten globale Optima bestimmt werden, ohne dass der Anwender ein tiefes Expertenwissen der mathematischen Optimierung einbringen muss. Für einige spezielle Szenarien der Mehrfahrzeug-Kooperation und -Koordination wurden in den vergangenen Jahren MILP-basierte Lösungsansätze vorgestellt. So wurde in [RH02] für autonome Fahrzeuge und in [ED07] für kooperative Robotersysteme ein effizienter Weg der MILP-basierten Aktionszuweisung gezeigt. [SFH06] stellte einen Ansatz vor, der in der Lage ist, auch Sichtlinien-Konnektivitäten zu berücksichtigen.

Die Arbeit [BM99] ist ein Ausgangspunkt zahlreicher Untersuchungen zur approximativen Optimalsteuerung hybrider dynamischer Systeme mit gemischt-ganzzahligen linearen Approximationen. Zuverlässige Lösungsalgorithmen wurden kontinuierlich weiterentwickelt [BG06]. So konnte etwa innerhalb eines modellprädiktiven Ansatzes durch Wahl von Linearisierungspunkten zur Laufzeit und durch eine parallel laufende Simulation des nichtlinearen diskret-kontinuierlichen Originalproblems [SE02] die Genauigkeit und Effizienz im praktischen Einsatz verbessert werden.

Trotz einer hohen Zahl an vorgestellten Arbeiten zur Aufgabenzuweisung und Trajektorienplanung in Mehrfahrzeugsystemen sowie zu hybriden dynamischen Systemen lässt sich feststellen, dass für eine Reihe von Fragen und Problemstellungen nach wie vor keine allgemein gültigen und akzeptierten Antworten gegeben wurden:

- Bisher entwickelte Methoden beruhen meist auf Lösungen spezieller Detailfragen, die nicht in eine allgemeine Systematik der Planung kooperativer Mobilität eingebettet sind.
- Insbesondere gibt es bislang keine systematische Methodik, die – in Abwägung von

Genauigkeit und Effizienz – Abschätzungen der schwer entscheidbaren Fragen optimaler Steuerungen in kooperativen Mehrfahrzeugsystemen zulässt. Somit können grundlegende Fragen – wie etwa der Einfluss von spezifischen Kommunikationsanforderungen auf eine optimale kooperative Bewegung – bislang nicht oder nur in sehr speziellen Fällen befriedigend beantwortet werden.

- Die Lösung des Kernproblems kooperativer Mobilität mit nichtlinearer Bewegungsdynamik der einzelnen Fahrzeuge wird bislang nur mit starken Idealisierungen oder durch eine diskret-kontinuierliche Entkopplung angegangen. Die Aufgabenzuweisung erfolgt damit in der Regel auf Basis von Heuristiken.
- Aus Sicht der Anwender und Systementwickler mangelt es an einer Methodik, die es erlaubt, die Güte einer dezentralen Verhaltenssteuerung gegenüber einer global optimalen Referenzlösung abzuschätzen.

1.3 Ziele und wissenschaftliche Beiträge dieser Arbeit

Ziel der vorliegenden Arbeit ist die Entwicklung eines umfassenden Konzeptes zur Modellierung, Approximation und Optimierung, welches in der Lage ist, die enge Kopplung von diskreter Struktur und kontinuierlicher Bewegungsdynamik als Kernproblem kooperativer Mobilität zu berücksichtigen. Ein kombinatorischer Systemcharakter in Wechselwirkung mit der Hochdimensionalität eines nichtlinearen Optimalsteuerungsproblems stellt die grundsätzliche Schwierigkeit für eine effiziente Planungsmethodik dar.

Konkret berücksichtigen die Methoden der vorliegenden Arbeit dazu folgende Anforderungen:

- Der *diskret-kontinuierliche physikalische Problemcharakter* soll möglichst in allen vorgenommenen Approximationsschritten erhalten bleiben und somit eine simultane Festlegung optimaler Schaltsequenzen und Bewegungstrajektorien ermöglichen.
- Durch ein *flexibles Bewertungskonzept* soll es dem Anwender im Rahmen der mathematischen Optimierung möglich sein, sowohl eine Bewertung eintretender – diskreter oder kontinuierlicher – Systemzustände als auch deren zeitlicher Entwicklung vorzunehmen.
- *Modularität*: In Abhängigkeit der konkreten kooperativen Aufgaben soll es möglich sein, Modelle der Trajektorien- und Aufgabenplanung kooperierender Fahrzeuge um neue Aspekte zu erweitern oder zu reduzieren.
- *Skalierbarkeit* der Teilmodelle: Aufgrund der schweren Entscheidbarkeit diskret-kontinuierlicher Probleme ist es wünschenswert, die Approximationsgenauigkeit einzelner Modellkomponenten geeignet, z.B. über Einstellparameter, anpassen und so eine Abwägung von Effizienz und Genauigkeit vornehmen zu können.

In der Arbeit werden dazu in sich geschlossene Konzepte entwickelt, die auf der Theorie hybrider dynamischer Systeme und Methoden mathematischen nichtlinearen gemischt-ganzzahligen Optimalsteuerung aufbauen.

Mit Hilfe hierarchischer hybrider Zustandsautomaten wird die physikalische diskret-kontinuierliche Systemcharakteristik im Modell erfasst und über Transformationen der

gemischt-ganzzahligen mathematischen Optimierung zugänglich gemacht. Dabei werden in Abwägung von Genauigkeit und Berechnungseffizienz insbesondere lineare Approximationen verwendet. Das Konzept erlaubt in seinen Teilschritten eine flexible Auslegung nach aufgabenspezifischen, verschiedenartigen Optimierungskriterien. Im Gegensatz zum Großteil heuristischer Ansätze bleibt hier der hybride Systemcharakter in allen Phasen der Beschreibung, der Approximationen und der Optimierung erhalten. Die Probleme der Aufgabenzuweisung und Trajektorienplanung werden somit simultan gelöst.

Die Lösung der entstehenden endlichdimensionalen, diskret-kontinuierlichen Optimierungsprobleme erlaubt für viele Fragestellungen kooperativen Bewegungsverhaltens eine effizient berechenbare Abschätzung und kann für moderate Problemgrößen zu einer echtzeitfähigen, modellprädiktiven Regelung der Aufgabenzuweisung verwendet werden.

Die Eignung auf gemischt-ganzzahliger linearer Optimierung beruhender Methoden für verschiedene Fahrzeug-, Aufgaben- und Optimierungsziele wird anhand neuer Fragestellungen und anhand anerkannter Benchmarkprobleme ausgelotet. Dabei werden erstmalig in einem Optimierungsansatz sowohl Aspekte der Drahtloskommunikation heterogener Fahrzeugflotten sowie wechselnde Dynamikmodi als auch Bedingungen einer statischen oder dynamischen Umgebung betrachtet.

Unter Berücksichtigung physikalischer Bewegungsdynamik und der charakteristischen diskret-kontinuierlichen Kopplung von Systemzuständen werden somit neue Abschätzungen zur Systemauslegung und zur Entwicklung von Reglerkonzepten für das Kernproblem kooperativer Mobilität ermöglicht.

2 Beschreibung und Analyse hybrider dynamischer Systeme

Kapitelinhalt:

Nach einem Überblick über die Theorie hybrider dynamischer Systeme (Abschnitt 2.1) und den Stand der Forschung mit Blick auf kooperative Mehrfahrzeugsysteme betrachtet Abschnitt 2.2.1 das Modellierungskonzept hybrider Automaten, für das insbesondere eine hierarchische Erweiterung vorgeschlagen wird. Als Vorbereitung einer neuartigen Anwendung auf das Kernproblem kooperativer Mobilität werden Spezialfälle linearer Systembeschreibungen in Abschnitt 2.3 vorgestellt. In Abschnitt 2.4 werden schließlich Anwendungen und interdisziplinäre Vereinbarkeit von Methoden der Systemanalyse sowie deren Kenngrößen diskutiert.

Systeme mit enger Kopplung von kontinuierlichen und diskreten Zustandsübergängen werden als hybride (dynamische) Systeme bezeichnet. Sie ergeben sich als Kombination von zeitgetriebenen und ereignisgetriebenen Modellen, welche in enger Wechselwirkung stehen. Hybride dynamische Systeme werden unter verschiedenen Gesichtspunkten insbesondere in der Informatik, der Regelungstheorie und der Mathematik seit knapp zwei Jahrzehnten mit wachsendem Interesse untersucht und können Antworten auf eine Vielzahl von Anwendungsproblemen geben.

Populäre anwendungsorientierte Beispiele, die als hybride dynamische Systeme modelliert werden und Kern einer allgemeineren Problemstellung darstellen, sind (vgl. Abb. 2.1):

- Tanksysteme, Ventile (Kernproblem der chemischen Verfahrenstechnik, z.B. [Old05]),
- Fahrzeuge mit Gangschaltung (Kernproblem geschalteter Dynamiken, z.B. [Ger05]),
- der springende Ball (Kernproblem von Dynamikwechseln nach Stößen) oder
- die Kombination von Tempomat und Sicherheitsabstand im PKW (Kernproblem der Integration verschiedener Regelungsziele und strategischer Bewegungsplanung).

Grundsätzlich werden diskrete Strukturen im Modell gewählt, wenn Unstetigkeiten im System auftreten; das heißt bei Vorgängen, die keine Formulierung mit nichtlinearen hinreichend glatten Ausdrücken erlauben, wie sie zur Anwendung von Algorithmen der nichtlinearen System- und Regelungstheorie notwendig wären.

Neben Modellen, die einen diskret-kontinuierlichen physikalischen Systemcharakter widerspiegeln, kann ein hybrides dynamisches Modell auch in Folge stückweiser Approximation von nichtlinearen Dynamiken entstehen (vgl. dazu Abschnitt 3.2 oder z. B. [ADG07]).

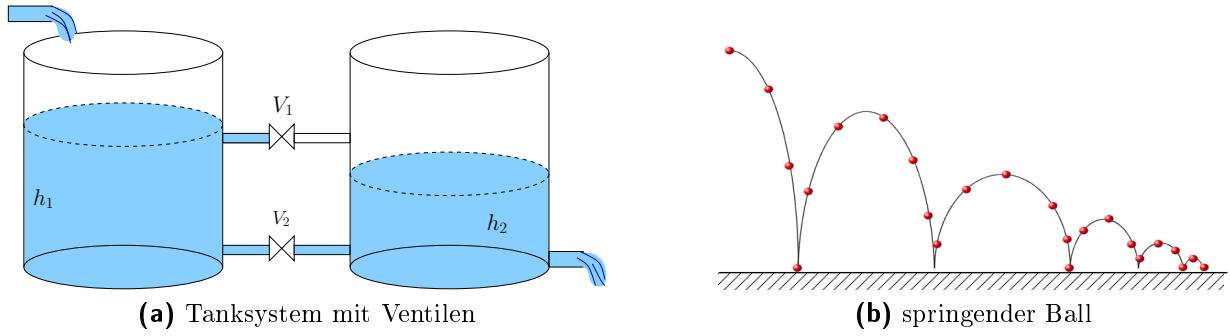


Abbildung 2.1: Beispiele diskret-kontinuierlicher Systemdynamik

2.1 Grundlegende Merkmale hybrider dynamischer Systeme

Hybride dynamische Systeme können zwischen verschiedenen Betriebszuständen umschalten, wobei jeder Modus durch seine eigenen charakteristischen dynamischen Regeln bestimmt ist. Solche Zustandswechsel können die Folge externer Steuersignale sein oder automatisch erfolgen, etwa nach Überschreiten einer Schranke durch eine Zustandsgröße oder nach Ablauf einer festen Zeitspanne [MB03]. Hybride Systeme besitzen sowohl kontinuierliche als auch diskrete Systemzustände. Grundlage allgemeiner Betrachtungen sei folgende Definition hybrider dynamischer Systeme (aus [CPPSV06]).

Definition 1. (hybrides dynamisches System)

Sei $\mathbf{u} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{U} \subset \mathbb{R}^{n_u}$ eine messbare Steuerfunktion, $\mathbf{z} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{V} \subset \mathbb{R}^{n_z}$ eine messbare Störfunktion und $\mathbb{X} \subset \mathbb{R}^{n_x}$ eine Menge. Es bezeichne $S_C(\mathbb{X}, \mathbb{U}, \mathbb{V})$ die Klasse zeitkontinuierlicher, dynamischer Systeme, welche durch die Gleichung

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{z}(t)) \quad (2.1)$$

definiert ist, mit $t \in \mathbb{R}_{\geq 0}$, $\mathbf{x}(t) \in \mathbb{X}$, so dass Gl. (2.1) für alle messbaren Steuer- und Störfunktionen $\mathbf{u}(t)$ bzw. $\mathbf{z}(t)$ bei gegebener Anfangsbedingung eine eindeutige Lösung $\mathbf{x}(t)$ besitzt.

Ein Tupel $H = (\mathbb{Q}, \mathbb{U}_D, \mathbb{E}, \mathbb{X}, \mathbb{U}, \mathbb{V}, S, \text{Inv}, R, G)$ heißt dann *hybrides System*, wobei

- \mathbb{Q} die Menge aller diskreten Zustände,
- \mathbb{U}_D die Menge diskreter Steuereingänge,
- $\mathbb{E} \subset \mathbb{Q} \times \mathbb{U}_D \times \mathbb{Q}$ die Menge diskreter Übergänge,
- \mathbb{X} die kontinuierliche Systemzustandsmenge,
- \mathbb{U} die Menge der kontinuierlichen Steuerungen,
- \mathbb{V} die Menge der kontinuierlichen Störgrößen,
- $\mathbf{f} : \mathbb{Q} \rightarrow S_C(\mathbb{X}, \mathbb{U}, \mathbb{V})$ eine Abbildung, die jedem diskreten Zustand ein zeitkontinuierliches dynamisches System zuordnet,

- $\mathbf{i} : \mathbb{Q} \rightarrow 2^{\mathbb{X} \times \mathbb{U}_D \times \mathbb{U} \times \mathbb{V}}$ eine Abbildung, die jedem diskreten Zustand eine Teilmengenbeschreibung zuordnet – genannt Invarianten –,
- $\mathbf{e} : \mathbb{E} \times \mathbb{X} \times \mathbb{U} \times \mathbb{V} \rightarrow 2^{\mathbb{X}}$ den Kanten eine Rückstellabbildung zuordnet,
- $\mathbf{j} : \mathbb{E} \rightarrow 2^{\mathbb{X} \times \mathbb{U} \times \mathbb{V}}$ eine Abbildung ist, die den Kanten Sprungbedingungen zuordnet.

Bemerkung 1. Auf ähnliche Weise lässt sich ein *zeitdiskretes* hybrides System definieren, indem die kontinuierliche Zeitvariable $t \in \mathbb{R}_{\geq 0}$ durch eine diskrete $k \in \mathbb{Z}_{\geq 0}$ ersetzt wird und in jedem Zustand diskret-dynamische Systeme betrachtet werden.

Diskrete Zustandsübergänge können als Folge eines diskreten Steuersignals oder auf Grund einer verletzten Invariante eintreten. Die Sprungbedingung muss dazu erfüllt sein. Die Abbildung R weist der Dynamik eines eintretenden diskreten Zustands Startwerte zu.

Hybride dynamische Systeme lassen sich noch allgemeiner definieren (z.B. [GJP08]). Im Lauf der Arbeit wird das jeweils betrachtete System mit weiteren Definitionen zur Modellierung (S. 14) und diskret-kontinuierlichen Optimalsteuerung (S. 28) präzisiert.

Wichtige Spezialfälle aus der Literatur lassen sich wie folgt klassifizieren:

- Geschaltete Systeme (intern / extern)
Systeme, die nur intern schalten, besitzen keine diskreten Steuereingänge. Diskrete Zustandsübergänge können von außen nur indirekt hervorgerufen werden.

Beispiel: In einem PKW mit Automatikgetriebe findet der Wechsel eines Ganges durch das Überschreiten einer bestimmten Drehzahl automatisch statt.
- Zeitkontinuierliche / -diskrete Probleme
Insbesondere in der digitalen Regelung zieht sich die Systembeschreibung auf Abtastraten gemessener Zustandsgrößen zurück. Rein analoge Regler folgen einer zeitkontinuierlichen Systembeschreibung.
- (Nicht-)Deterministische hybride Systeme
Ist das Eintreten diskreter Zustandsübergänge mit Wahrscheinlichkeiten beschrieben, oder weist die Rückstellabbildung Freiheitsgrade auf, spricht man von nichtdeterministischen Systemen.

Beispiel: Ein Fahrerassistenzsystem kann auf Basis von Sensordaten ein hybrides Modell aufbauen und Risiken von Überholmanövern abschätzen (z.B.[ASB07]).

Hybride dynamische Systeme werden insbesondere in der Informatik und in der Regelungstechnik betrachtet. Dies wird in Abschnitt 2.4 ausführlich diskutiert.

2.1.1 Logische Bedingungen

Ein wesentliches Merkmal der im Rahmen dieser Arbeit betrachteten kooperativen Mehrfahrzeugsysteme ist das Vorhandensein logischer Bedingungen, was sich im Systemmodell als diskrete Struktur wiederfindet.

Wir betrachten lediglich eine klassische zweiwertige Aussagenlogik. Die maßgebende Grundidee, logische Aussagen in ein mathematisches Modell aufzunehmen, welches der

logischer Ausdruck	lineare Ungleichung
$X_1 \vee X_2$	$\delta_1 + \delta_2 \geq 1$
$X_1 \wedge X_2$	$\delta_1 = 1, \delta_2 = 1$
\bar{X}_1	$1 - \delta_1 = 1$
$X_1 \Rightarrow X_2$	$\delta_1 - \delta_2 \leq 0$
$X_1 \Leftrightarrow X_2$	$\delta_1 - \delta_2 = 0$
$X_1 \oplus X_2$	$\delta_1 + \delta_2 = 1$

Tabelle 2.1: Algebraische Formulierung elementarer logischer Ausdrücke [WB96]

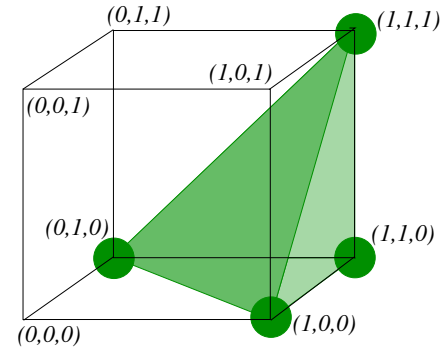


Abbildung 2.2: Geometrische Veranschaulichung: Die Ecken des Simplex beschreiben $(x_1 \wedge x_2) \vee (\bar{x}_1 \wedge x_2 \wedge \bar{x}_3) \vee (x_1 \wedge \bar{x}_2 \wedge \bar{x}_3)$

Optimierung zugänglich ist, besteht dabei darin, Boolesche Wahrheitswerte mit binären Variablen aus $\{0, 1\}$ zu formulieren und als Zahlenwerte zu identifizieren,

$$X \text{ ist wahr} \Leftrightarrow \delta = 1 \quad (2.2a)$$

$$X \text{ ist falsch} \Leftrightarrow \delta = 0. \quad (2.2b)$$

Eine Boolesche Variable X_n kann als Funktion weiterer Boolescher Variablen definiert sein,

$$X_n \Leftrightarrow f(X_1, X_2, \dots, X_{n-1}), \quad (2.3)$$

wobei f eine Kombination aus Verneinungen ($\bar{}$) und den Quantoren „und“ (\wedge) und „oder“ (\vee) ist. Alle diese logischen Ausdrücke (2.3) lassen sich in konjunktiver Normalform (KNF) notieren,

$$X_{n+1} = \bigwedge_{j=1}^k \left(\left(\bigvee_{i \in P_j} X_i \right) \vee \left(\bigvee_{i \in N_j} \bar{X}_i \right) \right), \quad N_j, P_j \subseteq \{1, \dots, n\}. \quad (2.4)$$

Neben der KNF ist für eine spätere Übersetzung in Beschränkungen an ein Optimierungsproblem auch eine minimale Darstellung logischer Ausdrücke von Interesse. Dies kann durch klassische Algorithmen der technischen Informatik hergestellt werden, etwa mittels Quine-McCluskey-Minimierung.

Auf Basis der Idee in Gl. (2.2) lassen sich rein logische Ausdrücke mit linearen Ungleichungen formulieren. Die KNF (2.4) resultiert damit etwa in

$$\begin{aligned} \delta_{n+1} &\leq \sum_{i \in P_1} \delta_i + \sum_{i \in N_1} (1 - \delta_i) \\ &\vdots \\ \delta_{n+1} &\leq \sum_{i \in P_k} \delta_i + \sum_{i \in N_k} (1 - \delta_i). \end{aligned}$$

Umformungen elementarer logischer Ausdrücke und die Transformation in einfache lineare Ungleichungen finden sich in Tabelle 2.1 dargestellt. Die Transformation logischer Aus-

drücke in eine Menge von Ungleichungen über binäre Variablen ist dabei nicht eindeutig. Beispielsweise gilt für die durch „ $X_3 \Leftrightarrow X_1 \wedge X_2$ “ beschriebene Menge:

$$\delta_3 = \delta_1 \cdot \delta_2 \text{ ist für } \delta_i \in \{0, 1\} \text{ äquivalent zu } \left\{ \begin{array}{l} -\delta_1 + \delta_3 \leq 0, \\ -\delta_2 + \delta_3 \leq 0, \\ \delta_1 + \delta_2 - \delta_3 \leq 0 \end{array} \right\}.$$

Die Frage nach der besten Formulierung kann oft nur in Einheit mit den sich anschließenden Methoden beantwortet werden. Die in der Tabelle angegebenen Formulierungen haben sich dabei in der Praxis häufig als gute Wahl bewährt. Eine geometrische Interpretation als Simplex (vgl. Abb. 2.2) steht im direkten Zusammenhang mit vorübergehenden Relaxierungen binärer Variablen zur Lösung diskret-kontinuierlicher Optimierungsprobleme (Abschnitt 3.1.3).

2.1.2 Gemischt-logische Bedingungen

Da im Rahmen dieser Arbeit der Aspekt enger Kopplung von diskreter Entscheidung und kontinuierlicher (Fahr-)Dynamik eine zentrale Rolle spielt, kommt der Formulierung gemischt-logischer Ausdrücke der Form

$$X \text{ ist wahr} \Rightarrow g(\mathbf{x}, \mathbf{u}) \leq 0 \quad (2.5)$$

eine besondere Bedeutung zu. Im Hinblick auf lineare Modelle hat sich in den vergangenen Jahren die „*Big-M*“-Methode [WB96] als wichtiges Modellierungswerkzeug behauptet. [BM99] verwendet den Ansatz bewusst im Kontext diskret-kontinuierlicher dynamischer Systeme und stellt – unter Berufung auf [WB96] – folgende Formulierungen vor:

$$[g(\mathbf{x}, \mathbf{u}) \leq 0] \Rightarrow \delta = 1 \quad \text{gilt genau dann, wenn} \quad g(\mathbf{x}, \mathbf{u}) \geq \varepsilon + (m - \varepsilon)\delta, \quad (2.6)$$

$$\delta = 1 \Rightarrow [g(\mathbf{x}, \mathbf{u}) \leq 0] \quad \text{gilt genau dann, wenn} \quad g(\mathbf{x}, \mathbf{u}) \leq M(1 - \delta). \quad (2.7)$$

Dabei ist für $(\mathbf{x}, \mathbf{u})^T \in \mathbb{X} \times \mathbb{U}$

$$M := \max_{(\mathbf{x}, \mathbf{u})^T \in \mathbb{X} \times \mathbb{U}} g(\mathbf{x}, \mathbf{u}) \quad m := \min_{(\mathbf{x}, \mathbf{u})^T \in \mathbb{X} \times \mathbb{U}} g(\mathbf{x}, \mathbf{u})$$

und ε ein kleiner Toleranzwert (typischerweise die Maschinengenauigkeit). Es sei erwähnt, dass die Schranken m und M theoretisch weniger streng gefasst werden können, es sich aber im Zuge der späteren Optimierung als deutlich effizienzverbessernd zeigt, die Schranken möglichst eng zu wählen. Tabelle 2.2 zeigt einige elementare Formulierungen.

Im Zusammenhang mit der Optimierung auf nichtlinearen Modellen spielt weiterhin die „*convex-hull*“-Formulierung eine wichtige Rolle. Diese wurde zur Beschreibung disjunkter Mengen im Kontext der gemischt-ganzzahligen nichtlinearen Optimierung entwickelt [Bal85] und insbesondere in [GL03] erweitert.

Anhand eines illustrierenden Beispiels stellen wir die Idee von *convex-hull* kurz der Big-M-Formulierung gegenüber und betrachten dazu die Formulierung (o.B.d.A: $\mathbf{0} \leq (\mathbf{x}, \mathbf{u})^T \leq (\mathbf{x}^{UB}, \mathbf{u}^{UB})^T$)¹,

$$\mathbf{g}_1(\mathbf{x}, \mathbf{u}) \leq 0 \oplus \mathbf{g}_2(\mathbf{x}, \mathbf{u}) \leq 0 \oplus \mathbf{g}_3(\mathbf{x}, \mathbf{u}) \leq 0. \quad (2.8)$$

¹Sofern nicht anders angegeben sind „<“- und „>“-Beziehungen stets komponentenweise zu lesen.

gemischt-logischer Ausdruck	äquivalente Big-M-Formulierung
$g(\mathbf{x}, \mathbf{u}) \leq 0 \vee \delta = 1$	$g(\mathbf{x}, \mathbf{u}) \leq \delta M$
$g(\mathbf{x}, \mathbf{u}) \leq 0 \wedge \delta = 1$	$g(\mathbf{x}, \mathbf{u}) - \delta \leq -1 + m(1 - \delta)$
$\delta = 1 \Rightarrow g(\mathbf{x}, \mathbf{u}) \leq 0$	$g(\mathbf{x}, \mathbf{u}) \leq (1 - \delta)M$
$\delta = 1 \Rightarrow 0 \leq g(\mathbf{x}, \mathbf{u})$	$(1 - \delta)m \leq g(\mathbf{x}, \mathbf{u})$
$g(\mathbf{x}, \mathbf{u}) \leq 0 \Rightarrow \delta = 1$	$g(\mathbf{x}, \mathbf{u}) \geq \varepsilon + (m - \varepsilon)\delta$

Tabelle 2.2: Wichtige Fälle der Big-M-Modellierung (vgl. auch [Glo05])

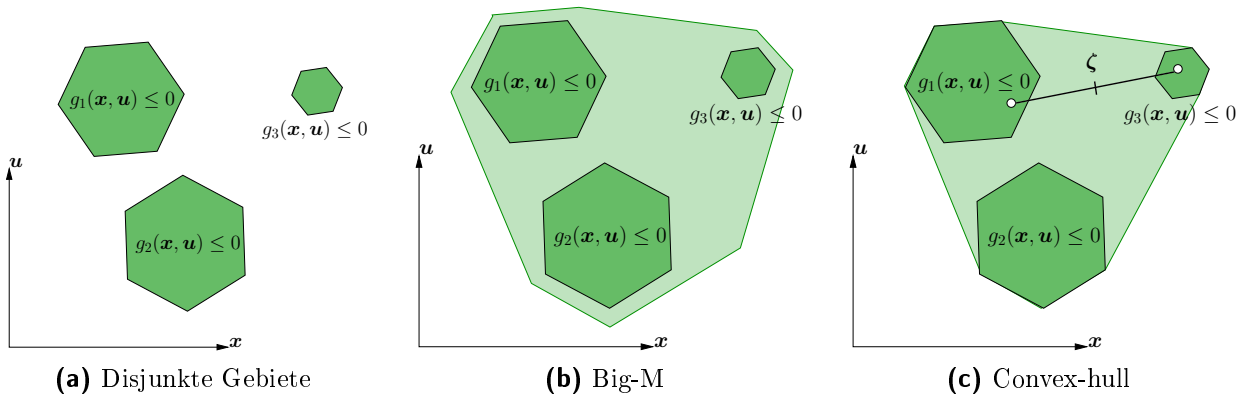


Abbildung 2.3: Relaxierungen mit Big-M und Convex-hull-Formulierung disjunkter Gebiete

Es ergeben sich dafür folgende Modellierungen für den in der Optimierung wichtigen relaxierten Fall, das heißt, die binären Variablen δ werden im Intervall $[0, 1]$ betrachtet:

Big-M:	Convex-hull:
$\mathbf{g}_i(\mathbf{x}, \mathbf{u}) \leq (1 - \delta_i) \mathbf{M}_i$	$\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2 + \boldsymbol{\zeta}_3 = \begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix}$
$\delta_1 + \delta_2 + \delta_3 = 1$	$\delta_1 + \delta_2 + \delta_3 = 1$
$\mathbf{M}_i \geq \max\{\mathbf{g}_i(\mathbf{x}, \mathbf{u})\}$	$\delta_i \mathbf{g}_i\left(\frac{\boldsymbol{\zeta}_i}{\delta_i}\right) \leq 0$
$i \in \{1, 2, 3\} \quad , \quad \delta_i \in [0, 1]$	$\mathbf{0} \leq \boldsymbol{\zeta}_i \leq \delta_i \begin{pmatrix} \mathbf{x}^{UB} \\ \mathbf{u}^{UB} \end{pmatrix}$
	$i \in \{1, 2, 3\} \quad , \quad \delta_i \in [0, 1]$

(2.9)

Da $\boldsymbol{\zeta}_i$ mindestens so schnell gegen Null strebt wie δ_i , existiert $\lim_{\delta_i \searrow 0} \|\mathbf{g}_i(\boldsymbol{\zeta}_i/\delta_i)\|$.

Sowohl die Big-M- als auch die convex-hull-Formulierung beschreibt bei konvexen Bereichen $\{(\mathbf{x}, \mathbf{u})^T \in \mathbb{X} \times \mathbb{U} \mid \mathbf{g}_i(\mathbf{x}, \mathbf{u}) \leq \mathbf{0}\}$ einen konvexen Bereich im Steuer-Zustands-Raum. Das convex-hull-Modell erhöht offensichtlich durch Einführung weiterer Hilfsgrößen die Zahl an Variablen der Modellbeschreibung. Sie besitzt allerdings die in der gemischt-ganzzahligen Optimierung sehr bedeutsame Eigenschaft, den zulässigen Suchraum sehr eng zu fassen (vgl. Abb. 2.3). Um dies auch für Big-M-Formulierungen zu gewährleisten, müssen die konstanten \mathbf{M}_i möglichst klein werden, was im Allgemeinen nur mit genauer Modellkenntnis bewerkstelligt werden kann.

Convex-hull-Methoden sind ein zentraler Ansatz in der *verallgemeinerten disjunktiven Programmierung (GDP)* und werden z.B. in [Old05] ausführlich diskutiert und angewendet.

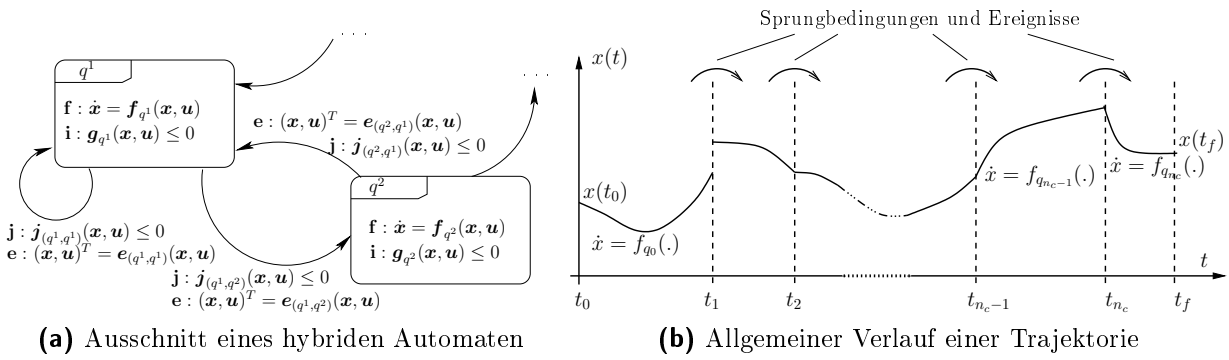


Abbildung 2.4: Hybrider Automat und allgemeine Gestalt einer beschriebenen kontinuierlichen Zustandstrajektorie. Die Länge und Abfolge der Phasen ist dabei nicht a priori festgelegt, sondern lediglich durch die Automatenstruktur eingeschränkt.

2.2 Modellierung hybrider dynamischer Systeme

Modellierungsansätze wurden im Lauf der Jahre begleitend zur analytischen Untersuchung hybrider dynamischer Systeme entwickelt. In [LBR97, LPS99] sind frühe Ansätze der formalen Beschreibung dargestellt, aus denen sich mittlerweile hybride Automaten und Petri-Netze [Pet62] als meist verwendete und untersuchte Modellierungsmethoden herausgebildet haben.

2.2.1 Hybride Zustandsautomaten

Betrachtet werden Mehrfahrzeugsysteme mit spezifischen Bewegungsmodi, deren zulässige Abfolgen Regeln unterliegen. Zur Beschreibung von Rollenverhalten ebenso wie von Rollen- und Aktionswechseln während der kooperativen Aufgabenerfüllung werden hybride Zustandsautomaten vorgeschlagen, die aus diskreten und kontinuierlichen Zuständen einer physikalischen Beschreibung bestehen und durch Bewegungsdifferentialgleichungen sowie durch algebraische Beschränkungen charakterisiert werden. Eine enge und formelle Kopplung diskreter und kontinuierlicher Zustandsdynamik in der Systembeschreibung wird damit berücksichtigt. Folgende Definition aus [Glo05] (aufbauend auf [Hen96]) bildet die Grundlage für die weiteren Betrachtungen.

Definition 2. (Hybrider Automat)

Ein *hybrider Zustandsautomat* $H = (\mathbb{Q}, \mathbb{E}, \mathbb{X}, \mathbb{U}, \mathbf{ini}, \mathbf{f}, \mathbf{j}, \mathbf{i}, \mathbf{e}, q_0)$ besteht aus

- einem endlichen gerichteten Multigraphen (\mathbb{Q}, \mathbb{E}) mit Knoten in \mathbb{Q} (so genannten Zuständen) und Kanten in $\mathbb{E} \subset \mathbb{Q} \times \mathbb{Q}$ (so genannten Schaltungen),
- einem Startzustand $q_0 \in \mathbb{Q}$,
- einer Menge kontinuierlicher Zustandsvariablen $\mathbb{X} \subset \mathbb{R}^{n_x}$,
- einer Menge kontinuierlicher Steuervariablen $\mathbb{U} \subset \mathbb{R}^{n_u}$,
- einer Abbildung \mathbf{ini} , die jeder Kante eine Anfangsbedingung zuweist,

- Invarianten, bereitgestellt durch die Abbildung \mathbf{i} , welche jedem Knoten eine zulässige Region der kontinuierlichen Zustands- und Steuervariablen mittels (Un-) Gleichungen zuweist,
- einer Abbildung \mathbf{f} , die jedem Zustand eine Flussgleichung zuordnet,
- einer Abbildung \mathbf{j} , die jeder Kante eine Sprungbedingung zuordnet und
- einer Abbildung \mathbf{e} , die den Kanten Ereignisse zuordnet, die beim Schalten eintreten.

Es wird angenommen, dass diskrete Zustandsübergänge in null Zeit stattfinden. Die Semantik hybrider Automaten kann darauf basierend über alternierende Sequenzen diskreter und kontinuierlicher Schritte definiert werden. In allgemeiner Form ergeben sich damit Zustandsverläufe, wie sie in Abbildung 2.4 veranschaulicht sind.

Hybride Zustandsautomaten sind im Zusammenhang mit der Beschreibung und Steuerung von Roboter- und Fahrzeugsystemen ein weit verbreitetes Hilfsmittel [ZKS03, CKC04, Glo05, BJSS07]. In den vergangenen Jahren haben sie sich zum häufigsten Modellierungswerkzeug hybrider Systeme entwickelt. Hybride Automaten stellen ein Bindeglied verschiedener Methoden der Systemtheorie, der regelungstechnischen Anwendung und der Informatik dar. Hybride Automaten existieren in verschiedenen „Dialekten“. Charakterisieren lassen sie sich unter anderem anhand einer gegebenenfalls vorhandenen Taktung und anhand des erlaubten Grades an Nichtlinearität der algebraischen Ausdrücke und Differentialgleichungen.

Wichtige Ausprägungen von Automaten, die im Rahmen dieser Arbeit berücksichtigt wurden, sind getaktete Automaten, bei denen Zustandsübergänge an festgelegten Zeitpunkten mit fester Taktrate stattfinden. Ferner betrachten wir im folgenden Abschnitt Erweiterungen auf hierarchische hybride Automaten.

2.2.2 Vorschlag zur Erweiterung auf Modelle mit hierarchischer Struktur

Die Erweiterung der im Abschnitt 2.2.1 vorgestellten Automaten um Hierarchien, Parallelität und Nebenläufigkeit ermöglicht es, dem hierarchischen Charakter der Relevanz von Verhaltensentscheidungen (Abb. 2.5a und z.B. [Ris09]), von Aufgabenklassen und der kooperativen Bewegungsplanung gerecht zu werden.

Dadurch werden verschiedene Abstraktions- und Idealisierungsebenen ermöglicht und es entsteht eine intuitive, übersichtlichere Modellform. Insbesondere im Fall einer hierarchischen Aufgabenstruktur mit zustandsabhängigen Dynamikmodi zeigen sich diese Vorteile. Die Betrachtung einer hierarchischen Beschreibung dynamischen Verhaltens schließt speziell auch die Beschreibung in verschiedenen Detaillierungs-, Abstraktions- und Approximationsgraden ein. Hierarchien wurden bereits von [ADE⁺01] vorgestellt und in [FMSS07] auf ein Szenario des Roboterfußballs angewandt. In Anlehnung daran werden nun die Komponenten *hierarchischer hybrider Automaten (HHA)* formal definiert.

Definition 3. (Hierarchischer hybrider Automat – HHA)

Seien \mathbb{X} , \mathbb{Q} , \mathbb{E} , \mathbb{U} wie in Definition 2. Folgende Struktur der Menge \mathbb{Q} definiert die Grundkomponenten eines *hierarchischen hybriden Automaten*:

Die endliche Menge diskreter Zustände \mathbb{Q} besteht aus den disjunkten Teilmengen $\mathbb{Q}_{\text{simple}}$, \mathbb{Q}_{comp} und \mathbb{Q}_{conc} – genannt einfache (simple), zusammengesetzte (composite)

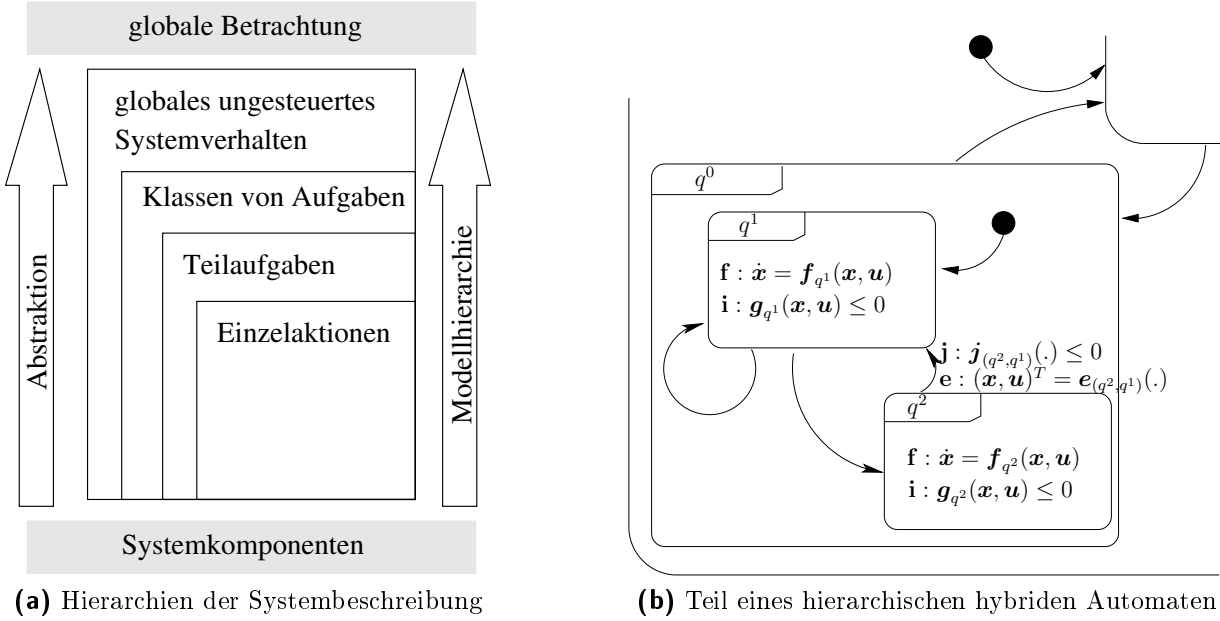


Abbildung 2.5: Die Modellierung mit hierarchischen hybriden Automaten entspricht der natürlichen Betrachtungsweise.

und nebenläufige (concurrent) Zustände, welche eine Menge ausgezeichneter *Startzustände* $q_0 \in \mathbb{Q}_{comp} \cup \mathbb{Q}_{conc}$ enthält.

Zum Zustand $q \in \mathbb{Q}$ gibt es keinen, einen oder mehrere *Anfangszustände* $\alpha(q)$: Ein einfacher Zustand hat null, ein zusammengesetzter Zustand genau einen und ein nebenläufiger Zustand mehr als einen Anfangszustand. Weiterhin besitzt jeder Zustand $q \in \mathbb{Q} \setminus \{q_0\}$ genau einen übergeordneten Zustand $\beta(q)$. Deshalb muss $\beta(q) \in \mathbb{Q}_{conc} \cup \mathbb{Q}_{comp}$ gelten. Ein nebenläufiger Zustand darf direkt keine anderen nebenläufigen Zustände enthalten. Weiterhin wird angenommen, dass alle Transitionen $(q_1, q_2) \in \mathbb{E}$ innerhalb einer Hierarchie stattfinden, d.h. $\beta(q_1) = \beta(q_2)$. Wir schreiben $\alpha^n(q)$ oder $\beta^n(q)$ für die n -fache Anwendung von α oder β auf q , insbesondere gilt $\alpha^0(q) = \beta^0(q) = q$.

Variablen $\mathbf{x} \in \mathbb{X}$ können lokal in einem bestimmten Zustand $\gamma(\mathbf{x}) \in \mathbb{Q}$ deklariert werden. Eine Variable $\mathbf{x} \in \mathbb{X}$ ist *gültig* in allen Zuständen $q \in \mathbb{Q}$ mit $\beta^n(q) = \gamma(\mathbf{x})$ für ein $n \geq 0$, außer eine andere Variable mit demselben Namen überschreibt sie lokal.

Die Begriffe *Flussbedingung* \mathbf{f} , *Invarianten* \mathbf{i} , *Sprungbedingungen* \mathbf{j} und *Ereignisse* \mathbf{e} gelten analog zu Definition 2.

Durch die Einführung von Hierarchien kann das Verhalten eines HHA nicht mehr als Sequenz von Zuständen beschrieben werden, sondern über eine Baumstruktur, die Konfiguration genannt wird. Eine formale Definition von Konfiguration und Semantik eines HHA ist in [FMSS07, RRSv08] beschrieben.

Im Falle nichtlinearen, insbesondere nicht-monotonen, Verlaufs kontinuierlicher Variablen treten in der Systemverifikation hybrider Automaten Entscheidungsprobleme auf.

2.2.3 Anwendung auf Mehrfahrzeugsysteme

Nachdem (hierarchische) hybride Automaten formal definiert wurden, werden diese nun mit konkreten Aspekten der Mehrfahrzeugkooperation identifiziert.

Eine Gruppe von n_v kooperierenden Fahrzeugen wird dazu im \mathbb{R}^2 in einer Zeitspanne $[t_0, t_f]$ betrachtet. Das System soll so modelliert werden, dass optimale Steuerungen im Weiteren berechnet werden können, welche unter Einhaltung der Bewegungsdifferentialgleichungen und physikalischen Beschränkungen ein gewünschtes Systemverhalten (z.B. Formationswechsel, taktische Kooperation) optimal bewerkstelligen.

Diskreter Automatenzustand: Die Knoten q eines hybriden Automaten markieren jeweils ein Bewegungsmuster, eine Rollenverteilung, eine konkrete (Teil-)Aufgabe und Forderungen nach Formationen. Ein diskreter Automatenzustand

$$q_k := q(t), \quad t_{k-1} < t \leq t_k, \quad k = 1, \dots, n_c$$

(mit Anfangszeit $t_0 = 0$ und Endzeit $t_{n_c} = t_f$), der den aktiven Knoten markiert, kann seinen Wert zu einem (variablen oder festen) Schaltzeitpunkt t_k ändern. Die Zahl der maximalen Schaltzeitpunkte n_c im System sei dabei als fest angenommen.

Ein Wechsel des diskreten Automatenzustands kann durch einen internen oder externen Schaltvorgang hervorgerufen werden.

Kontinuierlicher Systemzustand und Dynamik: Jedem Automatenzustand $q \in \mathbb{Q}$, also einer konkreten Systemaufgabe, wird durch die Abbildung „ \mathbf{f} “ eine Dynamik des Gesamtsystems

$$\dot{\mathbf{x}} := \begin{pmatrix} \dot{\mathbf{x}}^1 \\ \vdots \\ \dot{\mathbf{x}}^{n_v} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_q^1(\mathbf{x}^1(t), \mathbf{u}^1(t)) \\ \vdots \\ \mathbf{f}_q^{n_v}(\mathbf{x}^{n_v}(t), \mathbf{u}^{n_v}(t)) \end{pmatrix} = \mathbf{f}_q(\mathbf{x}, \mathbf{u}), \quad (2.10)$$

mit $(\mathbf{u}^1, \dots, \mathbf{u}^{n_v})^T =: \mathbf{u} \in \mathbb{U}$ zugeordnet. Dabei entspricht $\dot{\mathbf{x}}^i(t) = \mathbf{f}_q^i(\mathbf{x}^i(t), \mathbf{u}^i(t))$ der Bewegungsdynamik des Fahrzeugs i mit kontinuierlichem Systemzustand $\mathbf{x}^i(t) \in \mathbb{X} \subset \mathbb{R}^{n_{x,i}}$ (z.B. Position, Geschwindigkeit, Orientierung) und kontinuierlichen Steuervariablen $\mathbf{u}^i(t) \in \mathbb{U} \subset \mathbb{R}^{n_{u,i}}$ (z.B. Beschleunigung, Lenkwinkel). Abhängig vom Detaillierungsgrad des physikalischen Modells können diese Bewegungsdifferentialgleichungen von einfachen kinematischen bis hin zu komplexen nichtlinearen Fahrzeugdynamikmodellen reichen. Formal können Störgrößen $\mathbf{z}(t)$ über eine Erweiterung der Steuervariablen $\mathbf{u}(t)$ um Störkomponenten und die Anpassung der Bewegungsdifferentialgleichungen (2.10) berücksichtigt werden. Die Fahrzeugbewegung folgt genau dann der Gleichung (2.10), wenn q aktiv ist.

Neben steuerbaren Fahrzeugen kann das System noch weitere (reaktive, mobile) Objekte der Umgebung enthalten, wie zum Beispiel bewegliche Beobachtungsziele. Diese werden gegebenenfalls durch weitere Gleichungen der Form $\dot{\mathbf{x}} = \mathbf{f}_q(\mathbf{x})$ im Modell ergänzt.

Invarianten: Zustände und Steuervariablen der Fahrzeuge unterliegen Beschränkungen

$$\mathbf{g}_q(\mathbf{x}(t), \mathbf{u}(t)) \leq 0, \quad (2.11)$$

die sich meist direkt aus der Physik des Fahr- oder Flugzeugs (z.B. maximale Bewegungsradien, Höchstgeschwindigkeiten) und dessen Umgebung ergeben. Weitere Bedingungen ergeben sich durch Koordinations- und Kooperationsforderungen. Ferner werden durch Invarianten auch Grenzen und Eigenschaften des Operationsgebietes formuliert.

Sprungbedingungen und Ereignisse: Kanten stellen eine Veränderung diskreter Automatenzustände dar, wie sie zum Beispiel durch Kopplung, Massenverlust oder Aufgabenwechsel auftreten.

Sprungbedingungen definieren Voraussetzungen für Automaten-Zustandswechsel; so etwa die Forderung einer bestimmten räumlichen Anordnung, welche einen Formationswechsel einer Fahrzeugflotte erlaubt:

$$\mathbf{j}_{(q,q')}(\mathbf{x}(t_k - 0), \mathbf{u}(t_k - 0)) \leq \mathbf{0}, \quad (2.12)$$

mit $t \pm 0 := \lim_{\varepsilon \searrow 0} t \pm \varepsilon$. Die Übergänge von \mathbf{x} und \mathbf{u} folgen – sofern die der Kante $(q, q') \in \mathbb{E}$ zugeordneten Sprungbedingungen erfüllt sind – dem Ereignis

$$\begin{pmatrix} \mathbf{x}(t_k + 0) \\ \mathbf{u}(t_k + 0) \end{pmatrix} = \mathbf{e}_{(q,q')}(\mathbf{x}(t_k - 0), \mathbf{u}(t_k - 0)). \quad (2.13)$$

Es lassen sich auf diese Weise abrupte Änderungen kontinuierlicher Zustandsgrößen elegant modellieren; beispielsweise die Geschwindigkeitsänderung in Folge einer Kollision. In den meisten Fällen beschreiben Ereignisse (beziehungsweise Rückstellabbildungen, Definition 1) einfache Stetigkeitsforderungen.

Anfangs- und Endzustände: Diskrete und kontinuierliche Anfangszustände zur Startzeit t_0 sind in der Regel bekannt ($q(t_0) = q_0$, $\mathbf{x}^i(t_0) = \mathbf{x}_0^i$), können aber im Allgemeinen genauso wie die Endzeit t_f und die Endzustände frei oder fest gewählt sein. Die Festlegung einer verbindlichen Endzustandsmenge kann zur Formulierung einer kooperativen Aufgabe herangezogen werden.

Modellhierarchien: Als Inhalte oberer Modellhierarchien betrachten wir die taktische Entscheidung bezüglich der Regelungsziele, approximierende Systemdynamiken und globale Invarianten. In unteren Schichten finden sich Teilaufgaben feiner modelliert, ein genaueres Systemdynamikmodell und reaktive Reglerstrukturen („Reflexreaktionen“).

Im Beispiel des Roboterfußballs können in einer obersten Modellebene allgemeine Beschränkungen, etwa der Spielfeldgröße oder Kollisionsvermeidung formuliert werden. Eine weitere, darunter liegende, Ebene unterscheidet nach Angriffs- und Verteidigungsspiel, was beispielsweise eine spezifische Modellierung der Gegenspieler erlaubt. Tiefere Schichten modellieren einzelne Aktionen, wie etwa das Dribbeln eines Balls oder das Durchführen eines Pass-Spieles. Weitere darunter liegende Modellhierarchien können detailliertere Dynamikmodelle und Gegnerbeschreibungen enthalten.

2.3 Zeitdiskrete lineare hybride Systeme

Dieser Abschnitt beschränkt sich auf eine regelungstechnische Sicht und stellt zentrale Zusammenhänge vor. Fragen der Erreichbarkeit und Verifikation linearer hybrider Systeme werden in Abschnitt 2.4 allgemeiner diskutiert.

Falls die Systemdynamik (2.10) in der Form

$$\dot{\mathbf{x}}(t) = \mathbf{A}_q \mathbf{x}(t) + \mathbf{B}_q \mathbf{u}(t) + \mathbf{a}_q \quad (2.14a)$$

$$\mathbf{y}(t) = \mathbf{C}_q \mathbf{x}(t) \quad (2.14b)$$

mit zeitinvarianten Matrizen \mathbf{A}_q , \mathbf{B}_q , \mathbf{C}_q und Vektor \mathbf{a}_q geschrieben werden kann, liegt ein sogenanntes *affines, zeitinvariantes Zustandsraummodell* vor. \mathbf{y} beschreibt dabei in Gleichung (2.14b) den beobachtbaren *Systemausgang*.

Analog der Herleitung in [Hod07], basierend auf [Lun06], ergibt sich bei Anwendung auf das affine System (2.14) das *affine, zeitdiskrete Zustandsraummodell* ($k := t_k$)

$$\mathbf{x}(k+1) = \mathbf{\Phi}_q \mathbf{x}(k) + \mathbf{H}_q \mathbf{u}(k) + \mathbf{\phi}_q \quad (2.15a)$$

$$\mathbf{y}(k) = \mathbf{C}_q \mathbf{x}(k) \quad (2.15b)$$

mit zeitinvarianten Matrizen $\mathbf{\Phi}_q$, \mathbf{H}_q , \mathbf{C}_q und Vektor $\mathbf{\phi}_q$. Dabei ist $\mathbf{x}_k := \mathbf{x}(t_k)$ der kontinuierliche Systemzustand an diskreten Zeitpunkten $t_k \in \{t_0, t_1, \dots, t_{n_c}\}$ ($n_c \in \mathbb{N}$). Bei konstanter Abtastrate $\Delta_t := t_k - t_{k-1}$ ($\forall k \in \{1, 2, \dots, n_c\}$) kann die Darstellung (2.15) leicht mit den Berechnungsvorschriften [Lun06]

$$\mathbf{\Phi}_q = e^{\mathbf{A}_q \Delta_t}, \quad \mathbf{H}_q = \int_0^{\Delta_t} e^{\mathbf{A}_q \nu} d\nu \cdot \mathbf{B}_q \quad \text{und} \quad \mathbf{\phi}_q = \int_0^{\Delta_t} e^{-\mathbf{A}_q \nu} d\nu \cdot \mathbf{a}_q$$

aus der Darstellung (2.14) gewonnen werden. Falls $\mathbf{u}(t)$ in Gl. (2.14a) als stückweise konstant auf dem Gitter $(t_k)_{k=0, \dots, n_c}$ angenommen wird, liefert Gl. (2.15a) die exakten Werte für $\mathbf{x}(k+1)$.

2.3.1 Gemischt logisch-dynamische Beschreibung (MLD)

Gemischt logisch-dynamische Systeme (*mixed logical dynamical system, MLD-System*) stellen für einen Großteil wissenschaftlicher Untersuchungen im Bereich der zeitdiskreten, diskret-kontinuierlichen, linearen Systeme in den vergangenen Jahren den Ausgangspunkt der Systembeschreibung dar. MLD-Systeme sind eine verallgemeinerte Formulierung einer Vielzahl von Modellen, zu denen lineare hybride Systeme, endliche Zustandsautomaten und beschränkte, lineare Systeme gehören.

Als Vorbereitung einer neuartigen Anwendung auf das Kernproblem kooperativer Mobilität werden eine Definition und charakterisierende Merkmale von MLD-Systemen aus [BM99] vorgestellt.

Definition 4. (gemischt logisch-dynamisches System – MLD-System)

Eine Systembeschreibung folgender Form heißt *gemischt logisch-dynamische Systembeschreibung*:

$$\mathbf{x}(k+1) = \mathbf{A}_k \mathbf{x}(k) + \mathbf{B}_{1k} \mathbf{u}(k) + \mathbf{B}_{2k} \boldsymbol{\delta}(k) + \mathbf{B}_{3k} \mathbf{z}(k) \quad (2.16a)$$

$$\mathbf{y}(k) = \mathbf{C}_k \mathbf{x}(k) + \mathbf{D}_{1k} \mathbf{u}(k) + \mathbf{D}_{2k} \boldsymbol{\delta}(k) + \mathbf{D}_{3k} \mathbf{z}(k) \quad (2.16b)$$

$$\mathbf{E}_{2k} \boldsymbol{\delta}(k) + \mathbf{E}_{3k} \mathbf{z}(k) \leq \mathbf{E}_{1k} \mathbf{u}(k) + \mathbf{E}_{4k} \mathbf{x}(k) + \mathbf{E}_{5k} \quad (2.16c)$$

mit $k \in \{0, 1, 2, \dots\}$.

Der Systemzustand $\mathbf{x} = \begin{pmatrix} \mathbf{x}_c \\ \mathbf{x}_l \end{pmatrix} \in \mathbb{R}^{\tilde{n}_c} \times \{0, 1\}^{n_l}$ ($n := \tilde{n}_c + n_l$),

der Systemausgang $\mathbf{y} = \begin{pmatrix} \mathbf{y}_c \\ \mathbf{y}_l \end{pmatrix} \in \mathbb{R}^{p_c} \times \{0, 1\}^{p_l}$ ($p := p_c + p_l$) und

der Systemeingang $\mathbf{u} = \begin{pmatrix} \mathbf{u}_c \\ \mathbf{u}_l \end{pmatrix} \in \mathbb{R}^{m_c} \times \{0, 1\}^{m_l}$ ($m := m_c + m_l$)

besitzen dabei jeweils einen kontinuierlichen (Index c) und binärwertigen (Index l) Anteil. $\delta \in \{0, 1\}^{n_\delta}$ und $z \in \mathbb{R}^{n_z}$ bezeichnen binärwertige und kontinuierliche Hilfsvariablen.

In Bezug auf Definition 1 gilt $\{\mathbf{x}_l \mid \text{Ungl. (2.16c) ist erfüllbar}\} \cong \mathbb{Q}$ und $\{\mathbf{u}_l \mid \text{Ungl. (2.16c) ist erfüllbar}\} \cong \mathbb{U}_D$. In Bezug auf Definition 2 gilt $x_c \in \mathbb{X}$, $u_c \in \mathbb{U}$, $\{\mathbf{x}_l \mid \text{Ungl. (2.16c) ist erfüllbar}\} \cup \{\mathbf{u}_l \mid \text{Ungl. (2.16c) ist erfüllbar}\} \cong \mathbb{Q}$. Der Matrizen-Index k ist eine abkürzende Schreibweise für $\mathbf{A}(t_k) =: \mathbf{A}_k$ und weist darauf hin, dass sich der diskrete Systemzustand in jedem Takt ändern kann.

Grundsätzlich kann die Ungleichung (2.16c) für viele Werte von δ und/oder z erfüllt werden. Allerdings wünscht man sich andererseits, dass $\mathbf{x}(k+1)$, $\mathbf{y}(k)$ eindeutig durch $\mathbf{x}(k)$ und $\mathbf{u}(k)$ bestimmt sind. Dies motiviert folgende Definition.

Definition 5. (well-posed-Eigenschaft, [BM99])

Es bezeichne $[\mathbf{B}_{2k}]^i$ die i -te Spalte der Matrix \mathbf{B}_{2k} . Sei $\mathbf{I}_{Bk} := \{i \mid [\mathbf{B}_{2k}]^i \neq \mathbf{0}\}$ definiert als die Menge aller Indizes der von $\mathbf{0}$ verschiedenen Spalten von \mathbf{B}_{2k} . Analog seien \mathbf{I}_{Dk} , \mathbf{J}_{Bk} und \mathbf{J}_{Dk} als Indexmengen der von $\mathbf{0}$ verschiedenen Spalten von \mathbf{D}_{2k} , \mathbf{B}_{3k} und \mathbf{D}_{3k} definiert.

Sei $\mathbf{I}_k := \mathbf{I}_{Bk} \cup \mathbf{I}_{Dk}$, $\mathbf{J}_k := \mathbf{J}_{Bk} \cup \mathbf{J}_{Dk}$.

Ein MLD-System (2.16) heißt *well-posed* genau dann, wenn $\forall k \in \mathbb{Z}$ gilt:

- (i) $\exists \delta \in \{0, 1\}^{n_\delta} \exists z \in \mathbb{R}^{n_z} : \mathbf{x}(k), \mathbf{u}(k)$ erfüllen Gl. (2.16c) und $\mathbf{x}_l(k+1) \in \{0, 1\}^{n_l}$;
- (ii) $\forall i \in \mathbf{I}_k$ existiert eine Abbildung $\mathcal{D}_{ik} : \mathbb{R}^{n+m} \rightarrow \{0, 1\}$, $\mathcal{D}_{ik}(\mathbf{x}(k), \mathbf{u}(k)) = \delta_i(k)$ (i -te Komponente von δ);
- (iii) $\forall j \in \mathbf{J}_k$ existiert eine Abbildung $\mathcal{Z}_{jk} : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$, $\mathcal{Z}_{jk}(\mathbf{x}(k), \mathbf{u}(k)) = z_j(k)$ (j -te Komponente von z).

Ein MLD-System (2.16) heißt *vollständig well-posed* genau dann, wenn zusätzlich $\forall k \in \mathbb{Z} : \mathbf{I}_k = \{1, \dots, n_\delta\} \wedge \mathbf{J}_k = \{1, \dots, n_z\}$ gilt.

Bemerkung 2. Die Funktionen \mathcal{D}_{ik} und \mathcal{Z}_{jk} sind durch die Ungleichung (2.16c) implizit definiert. Ferner sind diese Funktionen aufgrund der Ganzzahligkeitsforderung $\delta_i \in \{0, 1\}$ nichtlinear.

Im Folgenden werden die Variablen $\delta_i(k)$ ($z_j(k)$) als *well-posed* bezeichnet, falls $i \in \mathbf{I}_k$ ($j \in \mathbf{J}_k$), andernfalls heißen sie *indefinit*.

In [BM99] wird weiterhin beschrieben, wie sich stückweise lineare Systemdynamik, Zustands- und Steuerbeschränkungen, diskretwertige Systemeingänge und eine Automatenstruktur als MLD-System formulieren lassen.

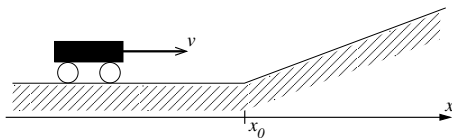


Abbildung 2.6: Fahrzeug an einer Rampe. Bei $x = x_0$ ändert sich die Fahrdynamik.

Beispielhaft wird die well-posed-Eigenschaft mit einem idealisierten Fahrzeug an einer Rampe illustriert (vgl. Abb. 2.6). Beschreibt man das Fahrzeug als Punktmasse x , das durch eine Geschwindigkeitsvorgabe u gesteuert wird, so berücksichtigt das Modell

$$x(k+1) = \begin{cases} x(k) + \Delta_t \cdot u(k) & , \text{ falls } x(k) < x_0 \\ x(k) + 0.7 \cdot \Delta_t \cdot u(k) & , \text{ falls } x(k) \geq x_0 \end{cases} \quad (2.17)$$

dass sich die Geschwindigkeit bei gleichbleibender Steuervorgabe verlangsamt, nachdem der Punkt x_0 passiert wurde.

Eine Übersetzung der Ausdrücke $[x(k) - x_0 > 0 \Leftrightarrow \delta = 1]$, $[\delta = 0 \Rightarrow z(k) = x(k) + \Delta_t u(k)]$ und $[\delta = 1 \Rightarrow z(k) = x(k) + 0.7 \Delta_t u(k)]$ gemäß Tabelle 2.2 erlaubt folgende MLD-Beschreibung für $x \in [0, 10]$ und $u \in [-1, 1]$,

$$x(k+1) = 0 \cdot x(k) + 0 \cdot u(k) + 0 \cdot \delta(k) + 1 \cdot z(k) \quad (2.18a)$$

$$y(k) = 1 \cdot x(k) \quad (2.18b)$$

$$\begin{bmatrix} m_1 - \varepsilon \\ M_1 \\ M_2 \\ -m_2 \\ -M_3 \\ m_3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \delta(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} z(k) \leq \begin{bmatrix} 0 \\ 0 \\ 0.7 \Delta_t \\ -0.7 \Delta_t \\ \Delta_t \\ -\Delta_t \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} u(k) + \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} x(k) + \begin{bmatrix} -x_0 - \varepsilon \\ x_0 + M_1 \\ M_2 \\ -m_2 \\ 0 \\ 0 \\ 0 \\ 10 \\ 1 \\ 1 \end{bmatrix} \quad (2.18c)$$

Dabei werden die Konstanten wie folgt gewählt:

$m_1 := \min\{x - x_0\}$, $M_1 := \max\{x - x_0\}$, $m_2 := \min\{z - x - 0.7 \Delta_t u\}$, $M_2 := \max\{z - x - 0.7 \Delta_t u\}$, $m_3 := \min\{z - x - \Delta_t u\}$ und $M_3 := \max\{z - x - \Delta_t u\}$.

In dieser Form ist das Modell well-posed, da mit $x(k)$ und $u(k)$ die Werte für $\delta(k)$ und $z(k)$ eindeutig bestimmt sind. Würde man ferner statt Gl. (2.18b)

$$y(k) = 1 \cdot x(k) + 1 \cdot \delta(k)$$

schreiben, so wäre das MLD-System vollständig well-posed.

2.3.2 Äquivalente Alternativen zu MLD-Beschreibungen

Ebenfalls sehr verbreitet ist die Formulierung stückweise affiner Systeme (*piecewise affine systems – PWA-Systeme*). Nach [HDB01] ist diese den MLD-Systemen äquivalent, sofern die well-posed-Eigenschaften erfüllt sind. Dazu folgende Definitionen aus [MB03].

Definition 6. (Polyeder, Polygon)

Sei $r, p \in \mathbb{N}$, $\Gamma \in \mathbb{R}^{r \times p}$, $\gamma \in \mathbb{R}^r$, dann heißt die Menge $\Theta = \{\theta \in \mathbb{R}^p \mid \Gamma \theta \leq \gamma\}$ (*konvexes*) *Polyeder*. Konvexe, beschränkte Polyeder in der Ebene heißen auch *Polygon*.

Stückweise affine Systeme sind auf einer polyedrischen Partition des Steuer-Zustands-Raumes definiert.

Definition 7. (Stückweise affines System – PWA-System)

Sei $s, \tilde{n}_c, n_l, m_c, m_l \in \mathbb{N}$ und $n := \tilde{n}_c + n_l$, $m := m_c + m_l$. Sei $\{P_i\}_{i=1}^s$ eine polyedrische Partition eines abgeschlossenen und beschränkten Steuer-Zustands-Raumes $\mathbb{X} \times \mathbb{U} \subset \mathbb{R}^{n+m}$. Sei $\mathbf{x} \in \mathbb{R}^{\tilde{n}_c} \times \{0, 1\}^{n_l}$, $\mathbf{u} \in \mathbb{R}^{m_c} \times \{0, 1\}^{m_l}$.

Eine Systembeschreibung der Form

$$\forall i = 1, \dots, s : \begin{pmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{pmatrix} \in P_i \Rightarrow \mathbf{x}(k+1) = \mathbf{A}^i \mathbf{x}(k) + \mathbf{B}^i \mathbf{u}(k) + \mathbf{f}^i \quad (2.19)$$

heißt *stückweise affine Systembeschreibung*.

Falls in (2.19) $n_l = m_l = 0$ gilt und die Abbildung $(\mathbf{x}(k), \mathbf{u}(k)) \mapsto \mathbf{x}(k+1)$ stetig ist, wird das PWA-System als stetig bezeichnet.

Es existieren eine Reihe weiterer Beschreibungsformen, welche in ihrer zeitdiskreten Formulierung sowohl den PWA- als auch MLD-Systemen äquivalent sind. Auf Basis von [BFTM00, HDB01] werden in [MB03] „(erweiterte) linear komplementäre Systeme“ und „max-min-plus-scaling-Systeme“ genannt. Theoretische Eigenschaften lassen sich somit übertragen und Werkzeuge auf die verschiedenen Problembeschreibungen anwenden.

Während die MLD-Beschreibung so angelegt ist, dass sich aus der Systemmodellierung eine kompakte Formulierung ableiten lässt, ist die PWA-Beschreibung mehr durch die Anwendung selbst motiviert und folgt in meist intuitiver der physikalischen Modellierung.

Für das Beispiel aus Abb. 2.6 kann die Partition $\{[0, x_0) \times [-1, 1], [x_0, 10] \times [-1, 1]\}$ gewählt und mit Gl. (2.17) das PWA-Modell direkt formuliert werden.

2.4 Kenngrößen und Methoden der Analyse hybrider Systeme

Erste Untersuchungen zu hybriden dynamischen Systemen fanden weitestgehend parallel im Umfeld der Informatik einerseits sowie der Regelungstheorie und Mathematik andererseits statt. Seit einigen Jahren wird dazu verstärkt die Frage nach einer synergetischen Vereinbarkeit entwickelter Methoden diskutiert.

Im Zuge wissenschaftlicher Untersuchungen wurden mit Blick auf Systemanalysen und Reglerentwicklung verschiedene Werkzeuge vorgestellt, die beispielsweise in [SSKE01, CPPSV06] systematisch verglichen werden.

2.4.1 Methoden der Informatik

Die Entwicklung formaler Modelle und Methoden für autonome Mehrfahrzeugsysteme (Mehrrobotersysteme) geschieht mit dem primären Ziel, Werkzeuge zur Analyse bereitzustellen. Es wird vorrangig die formale Beschreibung und Verifikation hybrider Systeme untersucht. Dabei werden Eigenschaften betrachtet, die mit einer mathematischen Logik formal beschrieben werden können. Diese Ansätze arbeiten in der Regel auf einer hohen Abstraktionsebene und benutzen diskrete Modelle, beispielsweise logik-basierte Methoden [BG06, HW04], sequentielle Entscheidungsansätze, oder bedienen sich der Theorie diskreter Systeme. Üblicherweise werden in diesen Umgebungen Eigenschaften der zugrunde liegenden physikalischen Vorgänge vernachlässigt oder nur sehr grob approximiert, welche aus physikalischer Fahrzeugbewegung – etwa der Dynamik mobiler Roboter – und deren Interaktion mit der Umwelt bestehen. Inwieweit diese Approximationen tatsächlich zu einer zufriedenstellenden Analyse des Mehrrobotersystems führen, hängt sowohl von den Fahrzeugklassen als auch der Art der Aufgaben ab.

Die formale Analyse hybrider dynamischer Systeme ist aufgrund der Komplexität und Größenordnung der Modelle meist eine herausfordernde Aufgabe. Bei Verifikationsaufgaben wird dabei versucht, algorithmisch nach einer festen Anzahl von Schritten eine Antwort darauf zu geben, ob ein gegebenes hybrides System mit Anfangswerten bestimmte Eigenschaften erfüllt. Die Entscheidbarkeit spielt dabei eine wichtige Rolle und ist nur unter bestimm-

ten Voraussetzungen gegeben (vgl. als Überblick [AHLP00]). Für spezielle Modellklassen wurden verschiedene Verifikationswerkzeuge implementiert (siehe [SSKE01, CPPSV06]). Die Basis dieser Hilfsmittel sind Verfahren, welche die Menge aller möglichen Trajektorien exakt charakterisiert oder manipuliert, etwa mit Hilfe eines Computeralgebrasystems.

Mit Blick auf hybride Zustandsautomaten definiert [Hen96] lineare hybride Automaten, bei denen die Abbildungen \mathbf{ini} , \mathbf{f} und \mathbf{j} boolesche Kombinationen linearer Ungleichungen sein müssen und $\dot{\mathbf{x}}$ die einzige freie Variable der Flussgleichungen \mathbf{f} ist. Dieser Systemtyp erlaubt aussagekräftige Resultate, die zur Verifikation und Erreichbarkeitsanalyse, wie etwa in HyTech [HHWT95], herangezogen werden können.

Für den allgemeinen Fall nichtlinearer, hybrider, dynamischer Systeme sind diese Verifikationswerkzeuge nicht mehr verfügbar. Approximativ werden erreichbare Mengen, etwa mit Hilfe von Polyedern, in einer Reihe von Werkzeugen berechnet, so etwa in CHECKMATE [SRKC00] oder HYSDEL [TB04] in Kombination mit der MULTI-PARAMETRIC TOOLBOX – MPT [Kva08] (s.a. Anwendung in Abschnitt 5.3).

Das formale Repräsentieren von Mehrfahrzeugsystemen verlangt unter Umständen nach der Möglichkeit, Nebenläufigkeit und Parallelität im Verhalten im Rahmen einer hierarchischen Modellierung auszudrücken. In [FMSS07] wird eine Vorgehensweise beschrieben, welche eine Analyse von Mehrrobotersystem-Spezifikationen durch eine Übersetzung in einfache hybride Automaten und die Anwendung von Standard-Modelchecking-Werkzeugen zur Systemverifikation erlaubt.

CONSTRAINT LOGIC PROGRAMMING – CLP [MS98] ist ein Programmierungsparadigma, bei dem Relationen unter Variablen als „constraints“ ausgedrückt werden. CLP wurde bereits zur Modellierung hybrider Systeme mit Differentialgleichungen angewendet [HW04]. Eine hohe Effizienz darf dabei allerdings nur erwartet werden, wenn eine vollständige CLP-Sprache wie zum Beispiel ECLIPSE PROLOG [Krz07] verwendet wird, bei der mehrere Constraint-Löser verfügbar sind, einschließlich Lösern für Constraints über kontinuierliche und diskrete Variablen sowie Optimierungs-Constraints auf kontinuierlichen Variablen. CLP erlaubt die Behandlung von HHA Beschreibungen in PROLOG-Notation. Optimierung, Sprungbedingungen und Flüsse können durch Constraints ausgedrückt werden. Genauer wurde dieses Vorgehen in [RRSv08] beschrieben.

Neben der Betrachtung von Mehrfahrzeug- und Mehrrobotersystemen, ist die Untersuchung sequentieller Eigenschaften in allen Anwendungsgebieten hybrider dynamischer Systeme Gegenstand der Systemanalyse; so etwa in der modellbasierten Verifikation der Verfahrenstechnik [Stu00] oder in der Sicherheitsbewertung von Fahrstrategien auf Basis von Erreichbarkeitsanalysen [ASB07].

2.4.2 Methoden der Regelungstheorie

Die vorliegende Arbeit beschränkt sich darauf, zeitdiskrete, lineare hybride Systeme zu analysieren. In [Hod07] werden zeitdiskrete lineare Systeme (Gl. 2.15a) zur linearen Systemtheorie in Bezug gesetzt und festgestellt, dass sich trotz der theoretischen Nähe zur linearen Systemtheorie die Begriffe *Steuerbarkeit*, *Beobachtbarkeit* und *Stabilität* nicht ohne Weiteres auf affine zeitdiskrete Modelle übertragen lassen.

Für die *Steuerbarkeit* eines linearen Systems wird (z.B. in [Lun06]) gefordert, dass das System aus jedem beliebigen Anfangszustand \mathbf{x}_0 in endlicher Zeit in den Endzustand $\mathbf{x}_f = 0$ überführt werden kann. Diese Forderung ist gleichbedeutend damit, dass jeder beliebige Endzustand, auch wenn dieser keine Ruhelage des Systems darstellt, vom An-

fangszustand aus erreicht werden kann. Dabei macht diese Forderung keine Aussagen über den Verlauf der Trajektorie vom Anfangszustand zum Endzustand und auch nicht darüber, wie groß die Stellgrößen werden dürfen. Bei zustands- und steuerbeschränkten Systemen ist dies problematisch und man geht bei hybriden Systemen zum Begriff der Erreichbarkeit über.

Die *Beobachtbarkeit* eines linearen Systems ist nach [Lun06, BFTM00] gegeben, wenn der Anfangszustand aus bekannter Ausgangsgröße und bekannter Eingangsgröße rekonstruiert werden kann. Aufgrund der Zustands- und Steuerbeschränkung ist dabei nicht garantiert, dass die Trajektorie die Beschränkungen erfüllt. In [Cv04] wird deshalb gefordert, dass auch alle Ereignisse während des Trajektorienverlaufs bekannt sein müssen, damit eine Rekonstruktion des Anfangszustandes stattfinden kann.

Die Untersuchung von *Stabilität* bildet den Kern der regelungstechnischen Systemanalyse. Stabilität heißt, dass kleine Störungen im System nur kleine Abweichungen in der Systemantwort nach sich ziehen. Je nachdem, was als Störung und was als Systemantwort definiert ist, gibt es eine Vielzahl von Stabilitätsauffassungen.

Unterschieden wird nach Zustands- oder Übertragungsstabilität. Unter Zustandsstabilität, auch genannt Lyapunov-Stabilität, versteht man ein System, das nach Auslenkung eines Zustands aus einer Ruhelage wieder dieser Ruhelage zustrebt. Übertragungsstabilität, oder auch Eingangs-Ausgangs-Stabilität, bedeutet, dass auf jede beschränkte Eingangsgröße das System mit einer beschränkten Ausgangsgröße antwortet.

Für hybride Systeme kann die Zustandsstabilität direkt für einen Zielzustand der Trajektorie angegeben werden. Für den Verlauf der Trajektorie bis zum Erreichen des Endgebietes muss nicht zwingend stabiles Verhalten gefordert sein. [PW06] bildet eine Brücke von Erreichbarkeitsanalysen im Modelchecking zu Stabilitätsuntersuchungen hybrider Systeme. Stabilität ist hier nicht wie üblich um einen Gleichgewichtspunkt, sondern um ein Gebiet definiert. Von Stabilität wird konsequenterweise dann gesprochen, wenn diese Gebiete erreicht und nicht wieder verlassen werden.

Stabilität in MLD-Systemen: Für Systeme in MLD-Beschreibung werden in [BM99] zur Stabilität folgende Definitionen gegeben.

Definition 8. (Gleichgewichtspunkt und Stabilität für MLD-Systeme)

Ein Punkt $\mathbf{x}^* \in \mathbb{R}^{\tilde{n}_c} \times \{0, 1\}^{n_l}$ heißt *Gleichgewichtszustand* eines MLD-Systems (2.16) und der Steuerung $\mathbf{u}^* \in \mathbb{R}^{n_u} \times \{0, 1\}^{m_l}$, falls für $(\mathbf{x}^{*T}, \mathbf{u}^{*T})^T \in \mathbb{R}^{\tilde{n}_c} \times \{0, 1\}^{n_l} \times \mathbb{R}^{m_c} \times \{0, 1\}^{m_l}$ und $\forall k_0 \in \mathbb{N}$, $k \geq k_0$ gilt: $\mathbf{x}(k, k_0, \mathbf{x}^*, \mathbf{u}^*) = \mathbf{x}^*$. $(\mathbf{x}^*, \mathbf{u}^*)$ heißt *Gleichgewichtspaar*.

Dabei ist $\mathbf{x}(k, k_0, \mathbf{x}_0, \mathbf{u}^*)$ der Systemzustand, der sich durch $(k - k_0)$ -maliges Auswerten von (2.16) mit $\mathbf{u}(k) = \mathbf{u}^*$ und Startwert $\mathbf{x}(k_0) = \mathbf{x}_0$ ergibt. Ein Gleichgewichtspaar heißt *stabil* genau dann, wenn

$$\exists k_0 \in \mathbb{N}, \forall \varepsilon > 0, \exists \delta, \forall k \geq k_0, \forall \mathbf{x}_0 : \|\mathbf{x}_0 - \mathbf{x}^*\| \leq \delta \Rightarrow \|\mathbf{x}(k, k_0, \mathbf{x}_0, \mathbf{u}^*) - \mathbf{x}^*\| \leq \varepsilon.$$

Ein Gleichgewichtspaar heißt *asymptotisch stabil* genau dann, wenn \mathbf{x}^* stabil ist und $\exists r > 0$, so dass $\forall \mathbf{x}_0 \in \{\mathbf{x}_0 \mid \|\mathbf{x}_0 - \mathbf{x}^*\| \leq r\}$ und $\forall \varepsilon > 0$

$$\exists T \forall k \geq T : \|\mathbf{x}(k, k_0, \mathbf{x}^*, \mathbf{u}^*) - \mathbf{x}^*\| \leq \varepsilon.$$

Bemerkung 3. (nach [BM99]) Die asymptotische Konvergenz der binären Komponente $\mathbf{x}_l(k)$ nach \mathbf{x}_l^* ist äquivalent zur Existenz einer endlichen Zeit k^* , für die gilt

$\forall k \geq k^* : \mathbf{x}_l(k) \equiv \mathbf{x}_l^*$. Somit können durch Setzen von $\mathbf{x}_l = \mathbf{x}_l^*$ lokale Stabilitätseigenschaften für den kontinuierlichen Systemzustand \mathbf{x}_c formuliert werden. Es gibt weiterhin eine Umgebung \mathbf{x}_c^* in \mathbb{R}^{n_x} , innerhalb derer $\mathbf{x}_c^*(k)$ gestört werden kann und die Bewegungsgleichungen für $\mathbf{x}_l(k) = \mathbf{x}_l^*$ gültig bleiben.

Relevant zur Regelung hybrider Systeme sind vorrangig Methoden der optimierungs-basierten Steuerung, auf die wir in Kapitel 3 eingehen. Im Zusammenhang modellprädiktiver Regelungsstrategien wird in Abschnitt 3.3.3 Stabilität nochmals diskutiert.

Um Optimierungsmethoden auf die in 2.2.2 beschriebenen HHA-Modelle anwenden zu können, ist eine Übersetzung des HHA in ein System von (Un-)Gleichungen und Differentialgleichungen erforderlich, wie dies im Abschnitt 3.1.2 dargestellt wird. Für ein Beispiel der kooperativen Zielbeobachtung und des Roboterfußballs werden Ergebnisse dazu im Abschnitt 5.3 vorgestellt werden.

2.4.3 Integrationsvorschläge zu Ansätzen der Informatik und mathematischer Optimierung

Das Interesse der Informatik eher durch sequentielle Eigenschaften motiviert, so etwa in Erreichbarkeitsanalysen und Systemverifikation. Die Regelungstheorie orientiert sich mehr an Eigenschaften, die für die Steuerung und Regelung bedeutsam sind, so etwa Fragen der Stabilität, der System-Beobachtbarkeit und -Steuerbarkeit sowie der Optimalität. Ein gemeinsames Interesse findet sich in der Systemauslegung, beispielsweise in der Optimierung von System- und Regelparametern, und in der Simulation hybrider dynamischer Systeme.

Die Verwendung der Methoden aus Abschnitt 2.4.1 ermöglicht in erster Linie Erreichbarkeitsanalysen. Andererseits kann ein reiner diskret-kontinuierlicher Optimierungsansatz Zustandshierarchien und die allgemeine Modellstruktur erst nach einer Transformation in ein gemischt-ganzzahliges Modell darstellen und nicht direkt, wie beispielsweise im CLP-Ansatz mit hybriden Modelcheckern.

Neben den prinzipiellen Fragen, „ob“ (Erreichbarkeit) und „wie“ (Optimierung) ein hybrider Systemzustand am besten erreicht werden kann, gibt es Fragen, für die eine eindeutige methodische Zuordnung nur im Spezialfall möglich ist: „*Welches* ist der beste (oder schlechteste) erreichbare Zustand?“

Konkret hat CLP Stärken, wenn die logische, diskrete Systemstruktur entscheidenden Einfluss besitzt; die (gemischt-ganzzahlige) Optimierung arbeitet gerade dann effizient, wenn die kontinuierliche Dynamik für die gesuchte Antwort entscheidend ist. In Mehrfahrzeugsystemen ist es im Allgemeinen anwendungs- und situationsabhängig, welcher Aspekt die Antwort dominierend beeinflusst, so dass es konsequent ist, im konkreten Anwendungsfall die Methodenwahl abzuwägen und gegebenenfalls auch kombinierte Ansätze wie in [Mag05, RRSv08] auszuloten.

Aus dem Umfeld des Operations Research sind bereits kommerzielle Produkte (z.B. [IBM]) mit kombinierten Techniken der Logikprogrammierung und der gemischt-ganzzahligen Optimierung verfügbar. Auch [BG03, BG06] stellen effiziente kombinierte Ansätze dieser Art zur Untersuchung von Optimalsteuerungsproblemen in zeitdiskreten linearen hybriden Systemen vor. Die Grundidee besteht dabei darin, rein diskrete Problemanteile mit Hilfe der Logikprogrammierung zu lösen und eine MILP-Optimierung zu integrieren. Gegenüber einer reinen MILP-Optimierung konnte damit eine deutliche Effizienzsteigerung erzielt werden.

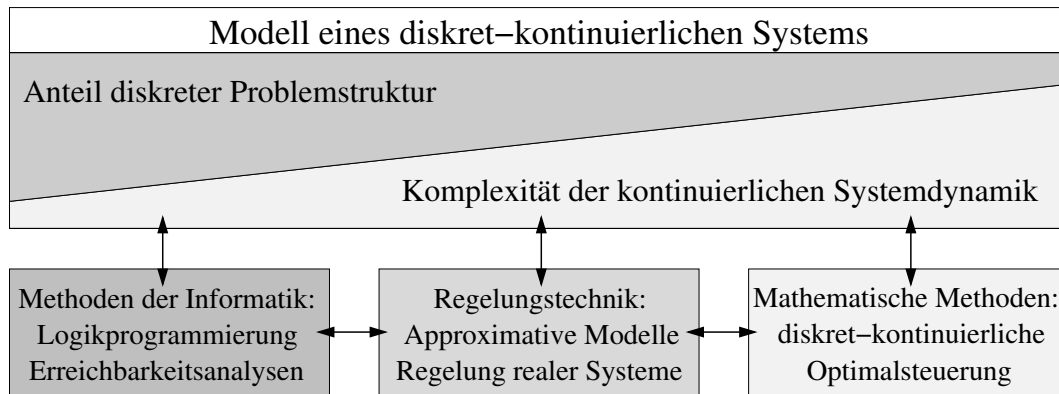


Abbildung 2.7: Je nach Modellstruktur zeigen verschiedene Ansätze ihre Leistungsfähigkeit. Die Methoden bieten dabei attraktive Kombinationsmöglichkeiten. Pfeile symbolisieren Aktivitäten aktueller Forschung und Entwicklung, wie sie auch in dieser Arbeit beleuchtet werden.

Im Rahmen des Modellierungskonzeptes für kooperative Fahrzeugsysteme in Abschnitt 4.1 werden wir den Einsatz von CLP-Methoden und Erreichbarkeitsanalyse als Komponente zur Verbesserung von Systembeschreibungen diskutieren.

Zusammenfassung des Kapitels

Mit Blick auf die systematische Modellierung, die Analyse und Steuerung kooperierender Mehrfahrzeuge stellen hybride dynamische Systeme den geeigneten Rahmen dar. Dabei bietet sich die Beschreibung mit Hilfe hybrider Zustandsautomaten an, zu denen eine hierarchische Erweiterung vorgeschlagen wurde. Vorbereitend für neuartige Anwendungen auf Probleme der kooperativen Mobilität wurden MLD-Beschreibungen und charakteristische Merkmale erläutert. Der aktuelle Stand der Forschung wurde aus dem Blickwinkel der Informatik und der Regelungstheorie betrachtet, interdisziplinäre Ansätze diskutiert und damit die Anwendung bestehender Techniken auf kooperative Mehrfahrzeuge motiviert.

Das Kapitel bildet somit insbesondere das Fundament für das Modellierungskonzept in Kapitel 4, sowie für die System- und Regleranalyse, wie sie in den Anwendungen des Abschnitts 5.3 vorgestellt wird.

3 Optimalsteuerung diskret-kontinuierlicher Systeme

Kapitelinhalt:

Die Inhalte bilden die Grundlage für die Optimierung auf diskret-kontinuierlichen Dynamikmodellen. Insbesondere werden Automatenmodelle der Optimierung zugänglich gemacht, ohne den diskreten und kontinuierlichen Problemanteil zu entkoppeln. Ein eigenes Konzept zur Modellapproximation entwickelt systematisch gemischt-ganzzahlig lineare Optimierungsprobleme, die zu numerischen Ansätzen der nichtlinearen diskret-kontinuierlichen Optimalsteuerung in Bezug gesetzt werden.

Darüber hinaus werden MILP-basierte MPC-Regler diskutiert und zur erstmaligen Anwendung auf das Kernproblem kooperativer Mobilität vorbereitet. Das Kapitel schließt mit einer Reihe weiterführender Konzepte der MPC-Anwendung auf nicht äquidistantem Gitter, der Dezentralisierung und der iterativen Verbesserung der Approximationsgenauigkeit.

In der Literatur finden sich verschiedene Bezeichnungen für die Problemklasse gemischt diskret-kontinuierlicher Optimalsteuerungsprobleme. So spricht man von „mixed-integer dynamic optimization“ oder „mixed-logic dynamic optimization“ (MIDO / MLDO, [Old05]), „mixed-integer optimal control problem“ (MIOCP) [Sag05] oder „hybrid optimal control“ [Sus99, BGH⁺02]. Im Rahmen dieser Arbeit verwenden wir den ebenfalls weit verbreiteten Begriff der *diskret-kontinuierlichen Optimalsteuerung* [Glo05], da er treffend die wesentliche Systemeigenschaft von enger Kopplung diskreter Entwicklung und kontinuierlicher Dynamik heraushebt.

3.1 Nichtlineare Systeme

3.1.1 Eigenschaften und Merkmale

Die enormen Verbesserungen von diskret-kontinuierlichen Optimierungsalgorithmen und von verfügbarer Rechnerleistung in der vergangenen Dekade machten die optimierungsbasierte Steuerung hybrider dynamischer Systeme für viele Bereiche der Forschung und Entwicklung attraktiv. Charakteristisch für die Optimierung hybrider dynamischer Systeme sind dynamische Prozesse mit wechselnden Betriebsmodi oder geschalteten (Un-)Gleichungsnebenbedingungen, die so geplant und gesteuert werden sollen, dass eine systemübergreifende Kostenfunktion minimiert wird.

Zunächst wird das gemischt diskret-kontinuierliche Optimalsteuerungsproblem als Basis für die weiteren Betrachtungen definiert.

Definition 9. (diskret-kontinuierliches Optimalsteuerungsproblem)

Es sei $\mathbb{X} \subset \mathbb{R}^{n_x}$, $\mathbb{U} \subset \mathbb{R}^{n_u}$ wie in Definition 2. Ferner sei $[t_0, t_f] \subset \mathbb{R}$ und $t_0 < t_1 < \dots < t_{n_c+1} = t_f$ definiere für $n_c \in \mathbb{N}$ eine Partition von $[t_0, t_f]$ in Teilintervalle $[t_k, t_{k+1}]$. Für $k = 0, \dots, n_c + 1$ sei $q_k \in \{q^1, \dots, q^{n_q}\} = \mathbb{Q}$ ein auf $[t_k, t_{k+1}]$ (bzw. $t_{n_c+1} = t_f$) gültiger diskreter Parameter.

Auf allen Teilintervallen $[t_k, t_{k+1}]$ seien die Funktionen $\mathbf{x} : [t_k, t_{k+1}] \rightarrow \mathbb{R}^{n_x}$, $\Phi_{q_{n_c+1}} : \mathbb{X} \rightarrow \mathbb{R}$, $\Phi_{q_k} : \mathbb{X} \times \mathbb{Q} \times \mathbb{X} \rightarrow \mathbb{R}$, $L_{q_k} : \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{R}$, $\mathbf{f}_{q_k} : \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{R}^{n_x}$, $\mathbf{g}_{q_k} : \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{R}^{n_{gq}}$, $\mathbf{h}_{q_k} : \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{R}^{n_{hq}}$, $\mathbf{r}_{q_k} : \mathbb{X} \rightarrow \mathbb{R}^{n_{rq}}$, $\mathbf{T} : \mathbb{X} \times \mathbb{Q} \times \mathbb{X} \rightarrow \mathbb{R}^{n_{Tq}}$ stetig differenzierbar.

Ferner sei $\mathbf{l} : \mathbb{R}^{n_c+1} \rightarrow \mathbb{R}$ linear und $\mathbf{u} : [t_k, t_{k+1}] \rightarrow \mathbb{R}^{n_u}$ stetig auf $[t_k, t_{k+1}]$. Mit

$$\begin{aligned} J[\mathbf{u}, q] := & \Phi_{q_{n_c+1}}(\mathbf{x}(t_f)) + \sum_{k=0}^{n_c} \Phi_{q_k}(\mathbf{x}(t_{k+1} - 0), q_{k+1}, \mathbf{x}(t_{k+1} + 0)) \\ & + \sum_{k=0}^{n_c} \int_{t_k}^{t_{k+1}} L_{q_k}(\mathbf{x}(t), \mathbf{u}(t)) dt \end{aligned} \quad (3.1a)$$

heißt das Minimierungsproblem

$$\min_{\mathbf{u}, q, t_f} J[\mathbf{u}, q] \quad (3.1b)$$

unter Berücksichtigung der Nebenbedingungen ($\forall k \in \{0, 1, \dots, n_c\}, \forall t \in [t_k, t_{k+1}]$)

$$\dot{\mathbf{x}}(t) = \mathbf{f}_{q_k}(\mathbf{x}(t), \mathbf{u}(t)) \quad (3.2a)$$

$$\mathbf{0} = \mathbf{g}_{q_k}(\mathbf{x}(t), \mathbf{u}(t)) \quad (3.2b)$$

$$\mathbf{0} \geq \mathbf{h}_{q_k}(\mathbf{x}(t), \mathbf{u}(t)) \quad (3.2c)$$

$$\mathbf{0} = \mathbf{r}_{q_k}(\mathbf{x}(t_k + 0)), \quad (3.2d)$$

$$\mathbf{0} \geq \mathbf{T}_{q_k}(\mathbf{x}(t_{k+1} - 0), q_{k+1}, \mathbf{x}(t_{k+1} + 0)) \quad (3.2e)$$

$$\mathbf{0} = \mathbf{r}_{q_{n_c+1}}(\mathbf{x}(t_f)), \quad (3.2f)$$

$$0 \geq \mathbf{l}(q_0, \dots, q_{n_c+1}) \quad (3.2g)$$

diskret-kontinuierliches Optimalsteuerungsproblem. Dabei heißen \mathbf{x} Zustandsvariablen, \mathbf{u} kontinuierliche Steuervariablen und q diskrete Steuervariablen. Die Schaltzeitpunkte t_k – insbesondere die Endzeit t_f – können fest vorgegeben oder frei wählbar sein. $(\mathbf{x}, \mathbf{u}, q_0, \dots, q_{n_c+1})$ heißt *zulässiger Steuerprozess*, falls die *Systemdynamik* (3.2a), die Gleichungsnebenbedingungen (3.2b), die Ungleichungsnebenbedingungen (3.2c), die Innere-Punkt-Bedingungen (3.2d, 3.2f) und die logischen Bedingungen (3.2g) erfüllt sind.

Ein zulässiger Steuerprozess $(\mathbf{x}^*, \mathbf{u}^*, q_0^*, \dots, q_{n_c+1}^*)$, der das Funktional (3.1) minimiert, heißt *optimal*.

Das Zielfunktional (3.1) heißt auch *Bolza-Funktional* und besteht aus zwei Termen. Der erste wird *Mayer-Term* genannt, der Integralteil *Lagrange-Term*.

Bemerkung 4. Notation: Für diskrete Variablen $q_k \in \mathbb{Q} = \{q^1, \dots, q^{n_q}\}$ werden tief gestellte Indizes verwendet, wenn die Zuordnung eines Wertes aus \mathbb{Q} zur Phase $[t_k, t_{k+1}]$ ausgedrückt werden soll. Die Elemente der Menge \mathbb{Q} werden mit hochgestelltem Index gezählt. „ q_k^s “ besagt also, dass $q(t)$ für $t \in [t_k, t_{k+1}]$ den Wert $q^s \in \mathbb{Q}$ annimmt.

Bemerkung 5. Diskrete Variablen q können sowohl *Steuer-* als auch *Zustandsvariablen* sein. Die passende Interpretation ist in den Freiheitsgraden der Nebenbedingungen (3.2e, 3.2g) für q und in dem Vorliegen deterministischer diskreter Übergänge begründet.

Ziel ist es, optimale Trajektorien zusammen mit zugehörigen Steuerungen des kooperativen Mehrfahrzeugsystems zu berechnen. Dabei sind Steuerungen als (abschnittsweise) stetige kontinuierliche Funktionen $\mathbf{u} : [t_0, t_f] \rightarrow \mathbb{R}^{n_u}$ und gegebenenfalls als diskrete Entscheidungsvariablen $q : [t_0, t_f] \rightarrow \{q^1, \dots, q^{n_q}\}$ zu unterscheiden. Im Kontext kooperativer Mehrfahrzeugsysteme stellen Steuerfunktionen \mathbf{u} Steuervorgaben an einzelne Fahrzeuge dar. Diskrete Entscheidungsvariablen modellieren die Zuweisung von Teilaufgaben.

Die physikalische Bewegung des Fahrzeugsystems beschreiben die Zustandsvariablen $\mathbf{x} : [t_0, t_f] \rightarrow \mathbb{R}^{n_x}$, deren Verlauf durch die Systemdynamik $\dot{\mathbf{x}} = \mathbf{f}_{q_k}(\mathbf{x}(t), \mathbf{u}(t))$ gegeben ist. Diese hängt von der aktuellen Konfiguration des Systems ab und kann im zeitlichen Verlauf aufgrund der Steuerungen $\mathbf{u}(t)$ variieren. Änderungen im Verlauf einer diskretwertigen Steuervariablen q , aber auch das Modellieren von Stoßvorgängen, kann zu unstetigen Änderungen der Zustandsvariablen zum Zeitpunkt t_k führen. Solche Sprünge können durch Transitionsbedingungen \mathbf{T}_{q_k} beschrieben werden.

Zustände und Steuerungen müssen während des zeitlichen Verlaufs verschiedene Nebenbedingungen einhalten. Dies können beispielsweise Beschränkungen der Fahrdynamik selbst (z.B. minimale Kurvenradien) oder auch Beschränkungen der Fahrzeugkoordination (z.B. Formationsvorgaben, Kollisionsverhinderung) sein. Es treten sowohl Gleichungs- als auch Ungleichungsnebenbedingungen auf, $0 = \mathbf{g}_{q_k}(\cdot)$ und $0 \leq \mathbf{h}_{q_k}(\cdot)$.

Beim Rundreiseproblem für Luftaufnahmen muss das Flugzeug vom Start bis zur Landung zu unbekannten Zeitpunkten t_k an jeweils einem der Wegpunkte sein. Solch eine Forderung an die Zustandsvariablen lässt sich durch Gleichungsbedingungen an (inneren) Zeitpunkten ($0 = \mathbf{r}_{q_k}(\cdot)$) fordern.

Um logische Zusammenhänge der diskreten Schaltungen und Parameter in die Problemformulierung mit aufzunehmen, werden bevorzugt lineare Bedingungen $0 \geq \mathbf{l}(q_0, \dots, q_{n_c+1})$ aufgestellt. Hier kann zum Beispiel beschrieben werden, dass nur höchstens eines der Fahrzeuge eine bestimmte Rolle übernehmen darf oder nur bestimmte Abfolgen an Teilaufgaben zulässig sind.

Die Güte einer Trajektorie und der dazugehörigen Steuerung wird schließlich durch das Zielfunktional $J[\mathbf{u}, q]$ gemessen.

Bemerkung 6. Das Zielfunktional (3.1) kann auch in reine Mayer-Form gebracht werden, indem eine zusätzliche Variable $x_L(t)$ mit Anfangswert $x_L(t_0) = 0$ und die zusätzliche Differentialgleichung

$$\dot{x}_L(t) = L(\mathbf{x}(t), \mathbf{u}(t))$$

eingeführt werden. Der Wert des Lagrangeterms entspricht somit dem Wert der Variablen $x_L(t_f)$. Auf diese Weise lassen sich auch Integralnebenbedingungen als zusätzliche Differentialgleichungen in das Optimalsteuerungsproblem einbetten.

Bemerkung 7. Das Optimalsteuerungsproblem wurde in Definition 9 in autonomer Form definiert, das heißt keine der vorkommenden Funktionen hängt explizit von der Zeit ab. Liegt ein Problem in nicht-autonomer Beschreibung vor, kann die Zeitvariable t durch eine Variable x_t mit Anfangswert $x(t_0) = 0$ und Dynamik $\dot{x}_t = 1$ ersetzt und somit das Problem in die autonome Formulierung überführt werden.

Bemerkung 8. Ein Optimalsteuerungsproblem mit freier Endzeit kann in ein äquivalentes Problem mit fester Anfangs- und Endzeit transformiert werden. Hierfür werden eine neue Zeitvariable $\tau \in [0, 1]$ und ein neuer Zustand x_{t_f} mit der Dynamik $\frac{dx_{t_f}}{d\tau} = 0$ eingeführt, so dass sich die tatsächliche Zeit mittels $t = t_0 + (x_{t_f} - t_0)\tau$ berechnen lässt. Die Endzeit entspricht dann dem Wert des neuen Zustands $x_{t_f} = t_f$.

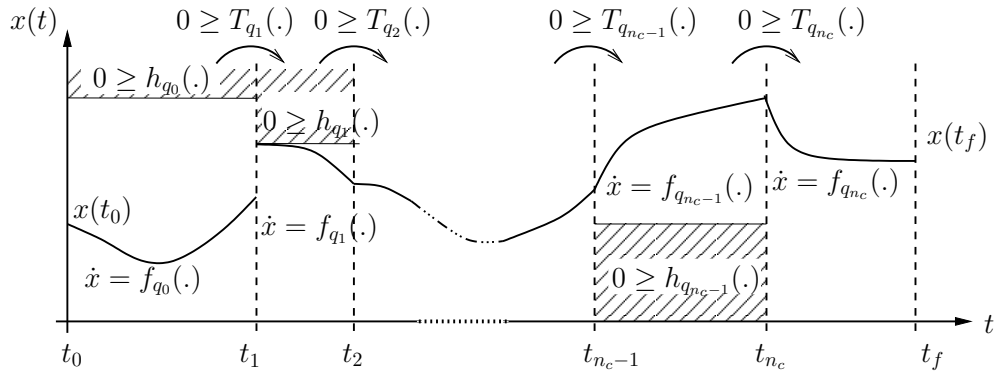


Abbildung 3.1: Kontinuierliche Zustandsgröße mit $n_c + 1$ Phasen, Übergänge treten an diskreten Schaltzeitpunkten t_k auf. Die Folge der q_k ist dabei vorab nicht vorgegeben. Gesucht ist der optimale Verlauf unter allen zulässigen Trajektorien

Bemerkung 9. In der Entwicklung numerischer Methoden für allgemeine nichtlineare diskret-kontinuierliche Optimalsteuerungsprobleme (z.B. [Glo05, Sag05]) werden meist detailliertere Definitionen verwendet. So wird dazu strikt nach Gleichungs- und Ungleichungsnebenbedingungen unterschieden, oder Systemparameter $\mathbf{p} \in \mathcal{D}_p \subset \mathbb{R}^{n_p}$ werden gesondert betrachtet. Diesen Fall behandeln wir als Spezialfall der Dynamik: $\dot{\mathbf{p}} = 0$.

Es werden lediglich deterministische Systeme betrachtet, für die gilt: Für jede zulässige Sequenz $(q_k)_{k=0}^{n_c+1}$ diskreter Systemzustände, wobei der Startzustand $\mathbf{x}(0)$ und auch der Steuerverlauf $\mathbf{u}(t)$, $0 \leq t \leq t_f$, gegeben sind, kann die Systemtrajektorie $\mathbf{x}(t)$, $0 \leq t \leq t_f$, aus Gleichung (3.2a), selbst unter Berücksichtigung von Sprung- und Schaltbedingungen (3.2e), eindeutig bestimmt werden.

Eine analytische Lösungsfindung für diskret-kontinuierliche Optimalsteuerungsprobleme ist – bis auf wenige sehr einfache Fälle – im Allgemeinen äußerst aufwändig oder gar unmöglich. Mit Blick auf praxisnahe Anwendungen beschränken wir uns deshalb auf numerische Lösungsmethoden.

Bemerkung 10. Da sich Systeme mit diskreten Steuervariablen in rein kontinuierliche Probleme transformieren lassen, sind theoretische Ergebnisse und Verfahren der kontinuierlichen Optimalsteuerung für diskret-kontinuierliche Probleme teilweise anwendbar. Zur Transformation verweist [Glo05] auf die Möglichkeit, die Diskretwertigkeit der Steuergrößen $q \in \{q^1, \dots, q^{n_q}\}$ über eine Gleichungsnebenbedingung

$$(q - q^1) \cdot (q - q^2) \cdot \dots \cdot (q - q^{n_q}) = 0$$

mit $q \in \mathbb{R}$ zu formulieren. Eine numerisch günstigere, approximative Variante wurde in [Ger05] vorgestellt.

Zwei fundamentalen Zusammenhänge der kontinuierlichen Optimalsteuerung werden hier lediglich benannt. Detaillierte Erweiterungen der vergangenen Jahre werden zum Beispiel in [SC07] zusammenfassend dargestellt, sind aber für den entwickelten Konzepte der vorliegenden Arbeit nicht relevant.

Pontryaginsches Maximumprinzip: Das Pontryaginsche Maximumprinzip (siehe Anhang Seite 107) gehört zu den bedeutendsten Sätzen der kontinuierlichen Optimal-

steuerung, auf das zahlreiche numerische Verfahren aufbauen. Eine Formulierung des Maximumprinzips für den vollen Umfang diskret-kontinuierlicher Optimalsteuerung findet sich erstmals in [Sus99]. Da es zu den hier entwickelten Methoden nur einen mittelbaren Bezug hat, verzichten wir auf die umfangreiche Darstellung dieser Zusammenhänge.

Wertefunktion und Hamilton-Jacobi-Bellman-Gleichung: Für alle betrachteten Problemstellungen gilt, dass Entscheidungen nur vom aktuellen Systemzustand und nicht von vergangenen Entscheidungen abhängen. Somit ist das Bellmansche Optimalitätsprinzip (vgl. Anhang Seite 108) anwendbar. Darauf basierend lassen sich Verfahren entwickeln, die rückwärts in der Zeit minimale Restkosten für konkrete Zustände berechnen. Dazu besitzt die Wertefunktion eine zentrale Bedeutung, mit der die sogenannte Hamilton-Jacobi-Bellmann-Gleichung (HJB, siehe Seite 108) formuliert wird.

3.1.2 Transformation eines Automatenmodells

Zur Berechnung optimierter Steuerungen auf Basis einer Modellbeschreibung durch hybride Automaten (vgl. Definition 2) wird eine Zeitstruktur $t_0 \leq t_1 \leq t_2 \leq \dots \leq t_{n_c+1} = t_f$ gemäß Definition 9 verwendet. Dabei sind t_k genau die Zeitpunkte, zu denen ein Wechsel der Automatenzustände $q \in \mathbb{Q} = \{q^1, \dots, q^{n_q}\}$ möglich ist. Automatenzustände entsprechen den diskreten Variablen aus Definition 9. Die Menge von Kanten \mathbb{E} eines hybriden Automaten lässt sich mit linearen Bedingungen der Gleichung (3.2g) beschreiben.

Die logische Verknüpfung der Nebenbedingungen kann wiederum durch den Big-M-Ansatz oder durch convex-hull-Relaxierungen erreicht werden. Welcher Ansatz im Speziellen besser geeignet ist, hängt u.a. vom Verhalten der Nebenbedingungen bei Relaxierung ab. Hier sind Nebenbedingungen, die sich aus der convex-hull-Relaxierung ergeben, strenger als die aus Big-M [GL03] (vgl. Abschnitt 2.1.2). Die Berücksichtigung spezieller Strukturen des Multigraphen kann darüber hinaus bei der Umwandlung des hybriden Automaten in ein diskret-kontinuierliches Optimalsteuerungsproblem helfen, unnötige Berechnungskomplexität zu vermeiden (siehe dazu auch [Glo05]).

Invarianten, welche jedem Knoten eine zulässige Region kontinuierlicher Zustands- und Steuervariablen zuordnen, finden sich in den (Un-)Gleichungen (3.2b) und (3.2c) wieder. Die Systemdynamik (3.2a) entspricht der Flussgleichung. Sprungbedingungen an diskrete Übergänge und zugehörige Ereignisse werden durch Gl. (3.2e) ausgedrückt.

Um die Kosten einer zeitlichen Entwicklung hybrider dynamischer Systeme zu messen, kann die Definition 2 hybrider Automaten um Lauf- und Schaltkosten erweitert werden. Schaltkosten $\Phi_{q_k}(\mathbf{x}(t_{k+1} - 0), q_{k+1}, \mathbf{x}(t_{k+1} + 0))$ werden dabei den Kanten und Laufkosten $L_{q_k}(\mathbf{x}(t), \mathbf{u}(t))$ den Automatenzuständen zugeordnet. Die Gesamtkosten der zeitlichen Entwicklung lassen sich dann bei gegebener Sequenz (q_k) , dem Steuerverlauf $\mathbf{u}(t)$ und den Anfangsbedingungen $\mathbf{x}(0) = \mathbf{x}_0$ mit Hilfe von Gleichung (3.1) berechnen.

Da im Vorfeld einer Optimierung nicht festgelegt werden soll, wie oft tatsächlich diskrete Zustandsänderungen auftreten, ist es gegebenenfalls nötig, alle Kanten der Form (q^j, q^j) in die Menge der möglichen Übergänge \mathbb{E} mit aufzunehmen. Alternativ schlägt [Glo05] für diese Problematik die Einführung *neutraler* Automatenzustände vor.

Die Optimalsteuerung auf nichtlinearen hybriden Automaten wurde bis vor wenigen Jahren kaum untersucht. Ansätze finden sich zum Beispiel in [RD05, CKC04]. Detailliertere Betrachtungen werden in [Glo05] dargelegt.

Entwicklung einer Transformation von Automatenhierarchien: Für die in Abschnitt 2.2.2 vorgestellte hierarchische Erweiterung von Automatenmodellen (vgl. Definition 3) können diskrete Zustände analog in lineare, algebraische Ausdrücke transformiert werden. Die Menge \mathbb{Q} wird dazu in die Teilmengen \mathbb{Q}_{simple} , \mathbb{Q}_{comp} und \mathbb{Q}_{conc} partitioniert.

Es seien b_q binäre Variablen, für die genau dann $b_q = 1$ gilt, wenn der Zustand $q \in \mathbb{Q}$ aktiv ist. Gilt zu jedem Schaltzeitpunkt t_k ($k \in \{0, 1, \dots, n_c\}$) mit $\mathbb{M}^* := \{q^i \in \mathbb{Q} \setminus \{q^*\} \mid q^* = \beta(q^i)\}$

$$\forall q_k^* \in \mathbb{Q}_{comp} \cup \mathbb{Q}_{conc} : b_{q_k^*} \geq \sum_{q_k^j \in \mathbb{M}^*} b_{q_k^j},$$

so bedeutet dies, dass genau dann ein Zustand derselben Hierarchie-Ebene aktiv sein muss, wenn deren übergeordneter Zustand q^* aktiv ist.

Die Zuordnungen \mathbf{f} , \mathbf{i} , \mathbf{j} und \mathbf{e} zum Automatenzustand $q \in \mathbb{Q}$ geschehen analog zum nicht-hierarchischem Fall. Dabei ist jedoch zu beachten, dass eine kontinuierliche Größe $\mathbf{x} \in \mathbb{X}$ in allen Zuständen $q \in \{q \in \mathbb{Q} \mid \exists n \geq 0 : \beta^n(q) = \gamma(\mathbf{x})\}$ gültig ist und jeweils die Systemdynamik $\dot{\mathbf{x}} = \mathbf{f}_q(\mathbf{x}, \mathbf{u})$ des aktiven Zustands der untersten Schicht bestimmend ist.

3.1.3 Numerische Verfahren

Die Grundidee der numerischen Lösung diskret-kontinuierlicher Optimalsteuerungsprobleme besteht darin, die Methoden der rein kontinuierlichen Optimalsteuerung mit Methoden der diskreten Optimierung in einem effizienten Algorithmus zu verzahnen. Die sich ergebende Berechnungskomplexität resultiert dabei einerseits aus dem kombinatorischen Problemcharakter des diskreten Modellanteils und andererseits aus der Hochdimensionalität eines zu lösenden kontinuierlichen Minimierungsproblems. [Sag05] weist darauf hin, dass diskret-kontinuierliche Optimalsteuerungsprobleme eine Verallgemeinerung gemischt-ganzzahliger nichtlinearer Optimierungsprobleme darstellen, die im Allgemeinen bereits \mathcal{NP} -schwer sind.

Im Rahmen der Arbeit dient die Lösung des diskret-kontinuierlichen Optimalsteuerungsproblems als Referenz für Lösungen auf linearisierte Approximationsmodellen. Die Basis allgemeiner Lösungsmethoden wird in knapper Form wiedergeben, um dann in Abschnitt 3.2 eigene Approximationsansätze zu positionieren und in Abschnitt 3.4 eine Zusammenführung mit approximativen Ansätzen vorzustellen.

Klassische Methoden der kontinuierlichen Optimalsteuerung: Im Wesentlichen lassen sich die existierenden Methoden in drei Ansätze unterteilen:

1. *indirekte Methoden:*

Notwendige Optimalitätsbedingungen (siehe Anhang S. 107) werden dazu verwendet, das Optimierungsproblem in ein Mehrpunkt-Randwertproblem zu übersetzen, welches dann zum Beispiel durch Mehrfachschießverfahren gelöst wird.

Die Hauptschwierigkeit indirekter Verfahren liegt in der numerisch stabil lösbaren Formulierung des Randwertproblems. Das bedeutet praktisch auch, dass nur kleine Modellveränderungen, etwa eine weitere Nebenbedingung, eine grundlegende Überarbeitung der Randwertproblemformulierung nach sich ziehen. Ferner ist das Bereitstellen von Startschätzungen, auch für adjungierte Variablen, eine unter Umständen

große Herausforderung an den Anwender. Für die indirekten Verfahren spricht hauptsächlich die hohe Genauigkeit, die sich damit erzielen lässt.

2. *direkte Methoden:*

Direkte Methoden basieren auf einer vollständigen Diskretisierung des Optimalsteuerungsproblems, beispielsweise durch parametrisierte stückweise Polynomfunktionen. Hierzu zählen Kollokationsverfahren (siehe unten), sowie Einfach- und Mehrfachschießverfahren. Aus deren Parametern wird ein endlichdimensionales nichtlineares Optimierungsproblem aufgestellt und mit entsprechenden Verfahren gelöst.

3. *dynamische Programmierung:*

Unter dynamischer Programmierung versteht man eine zeitdiskrete Technik, die sich auf das Bellmansche Optimalitätsprinzip (siehe S. 108) stützt. Sie steht im engen Zusammenhang mit der HJB, die sich ergibt, wenn die Zeitschritte gegen Null gehen.

Die Vorteile liegen in der globalen Optimalität der erzielten Lösungen. Diese entsteht durch die vollständige Untersuchung des Zustandsraumes, welche – offline durchgeführt und gespeichert – online über look-ups effizient realisiert werden kann. Die Hauptschwierigkeit ergibt sich bereits im rein kontinuierlichen Fall durch die hohe Dimension, in der eine partielle Differentialgleichung zu lösen ist. In der Literatur wird daher auch vom „Fluch der Dimensionalität“ („curse of dimensionality“) gesprochen.

Die Ausweitung der Methoden auf diskret-kontinuierliche Probleme findet sich in der Literatur in verschiedenen Abstufungen. So beschäftigen sich beispielsweise viele Arbeiten lediglich mit optimalen Schaltzeitpunkten linearer Systeme (z.B. [SBC06, DM06]) oder es werden Systeme untersucht, die eine Entkopplung von diskreter und kontinuierlicher Struktur zulassen. Ein systematischer, detaillierter Überblick findet sich etwa bei [Sag05, Glo05].

Wir stellen nun eine nichtlineare Methode zur numerischen Lösung allgemeiner diskret-kontinuierlicher Optimalsteuerungsprobleme vor, die weitestgehend auf [Glo05] beruht und in Kapitel 5 Referenzlösungen liefert.

Diskret-kontinuierliche direkte Kollokation: Dieser allgemeine nichtlineare numerische Lösungsansatz besteht aus einer Zerlegung des diskret-kontinuierlichen Optimalsteuerungsproblems in ein gekoppeltes diskret-dynamisches Optimierungsproblem mit äußerer und innerer Schleife (vgl. [BGH⁺02, vSG01]).

In der inneren Iteration werden dynamische Optimierungsprobleme betrachtet, bei denen die nichtlineare Zustandsdynamik in mehreren Phasen definiert ist (Abb. 3.1). Für jede Phase $[t_{k-1}, t_k]$ wird ein adaptives Zeitdiskretisierungsgitter eingeführt. Entlang dieses Zeitgitters werden die kontinuierlichen Zustandsvariablen $\mathbf{x}(t)$ und Steuervariablen $\mathbf{u}(t)$ approximiert [vSG01]. In Referenzlösungen der vorliegenden Arbeit werden für die Zustandsvariablen stetig differenzierbare stückweise kubische und für die Steuervariablen stetige stückweise lineare Polynome verwendet. Die Kollokationsbedingungen werden an Lobatto-Punkten, alle weiteren Nebenbedingungen auf den Punkten des Diskretisierungsgitters gefordert. Auf diese Weise wird das Optimalsteuerungsproblem mit (teilweise relaxierten) diskreten Variablen in ein großes, dünn besetztes, nichtlineares, beschränktes Optimierungsproblem transformiert, welches numerisch mit einem SQP-Verfahren¹ gelöst

¹Sequentielle quadratische Optimierung – sequential quadratic programming

wird. Ist eine Lösung gefunden, wird in Bereichen mit starker Verletzung der Kollokations- und Nebenbedingungen das Diskretisierungsgitter verfeinert.

In der äußeren Iteration wird eine Suche im diskreten Lösungsraum mit Branch-and-bound-Verfahren (B&B) durchgeführt. Im Allgemeinen geht durch eine vorübergehende Relaxierung diskreter Variablen an inneren Knoten des Suchbaums die physikalische Bedeutung verloren, was das Verfahren aber nicht einschränkt, sofern eine numerische Lösung existiert.

Die Effizienz der äußeren Iteration hängt maßgeblich von guten Startschätzungen sowie dem Bereitstellen guter unterer und oberer Schranken der Zielfunktion (3.1) ab [vSG01]. Um gute Startlösungen zu gewährleisten, kommen in den Arbeiten von [Glo05] Homotopieverfahren zum Einsatz, welche die benötigten Lösungen beim Durchlaufen des Suchbaums ineinander überführen.

Diskussion: Der vorgestellte numerische Ansatz ist auf eine sehr allgemeine Klasse nichtlinearer diskret-kontinuierlicher Probleme anwendbar. Sofern die Modelle moderaten Glattsheitsforderungen genügen, lassen sich Lösungen mit hoher messbarer Genauigkeit berechnen. Allgemeine diskret-kontinuierliche Optimalsteuerungsprobleme besitzen jedoch grundsätzliche Schwierigkeiten, die sich auf die vorgestellten Methoden übertragen:

Nichtlinearitäten verlangen eine ausreichend hohe Zahl an Kollokationspunkten, was insgesamt zu einer hohen Dimension des entstehenden nichtlinearen Optimierungsproblems (NLP) führt. Ferner bleibt die grundsätzliche NLP-Problematik erhalten, globale Optima zu finden.

Der diskrete Problemanteil, der als überlagerte kombinatorische Problemstruktur vieler NLPs in Erscheinung tritt, führt dazu, dass die Probleme mit wachsender Dimension eine exponentiell wachsende Komplexität zeigen.

Durch eine Diskretisierung auf ein gemischt-ganzzahliges NLP gelangt man zu \mathcal{NP} -schweren Problemen, deren Performanz im Wesentlichen durch Startschätzungen und durch untere Schranken für (relaxierte) Teilprobleme beeinflusst ist. Diese sind aber im Allgemeinen nur schwer zu generieren und verlangen vom Benutzer ein hohes Maß an Modellwissen und Erfahrung im Umgang mit diskret-kontinuierlichen Optimalsteuerungsproblemen.

3.2 Konzepte zur Modellapproximation

Die Modellapproximation begleitet den gesamten Prozess der Systemmodellierung, wie er in 4.1 und 4.2 nochmals genauer vorgestellt werden wird. Ziel der Approximation ist es dabei stets, die Effizienz der anschließenden Optimierung zu steigern, ohne dabei erforderliche Grenzen der Modellgenauigkeit zu verletzen.

Neben der separaten Effizienz-zu-Genauigkeit-Abwägung einzelner Modellierungsdetails ist ein insgesamt modularer Aufbau des Gesamtmodells mit skalierbarem Detaillierungsgrad der Teilmodelle äußerst zweckmäßig. In Anbetracht der resultierenden Problemkomplexität in Anwendung auf ein breites Spektrum verschiedenartiger Szenarien bleibt so eine flexible Handhabung der Konzepte gewährleistet.

Neben etablierten Linearisierungen stellt der folgende Abschnitt ein eigenes Konzept zur optimalen PWA-Hybridisierung mit minimalem Approximationsfehler vor.

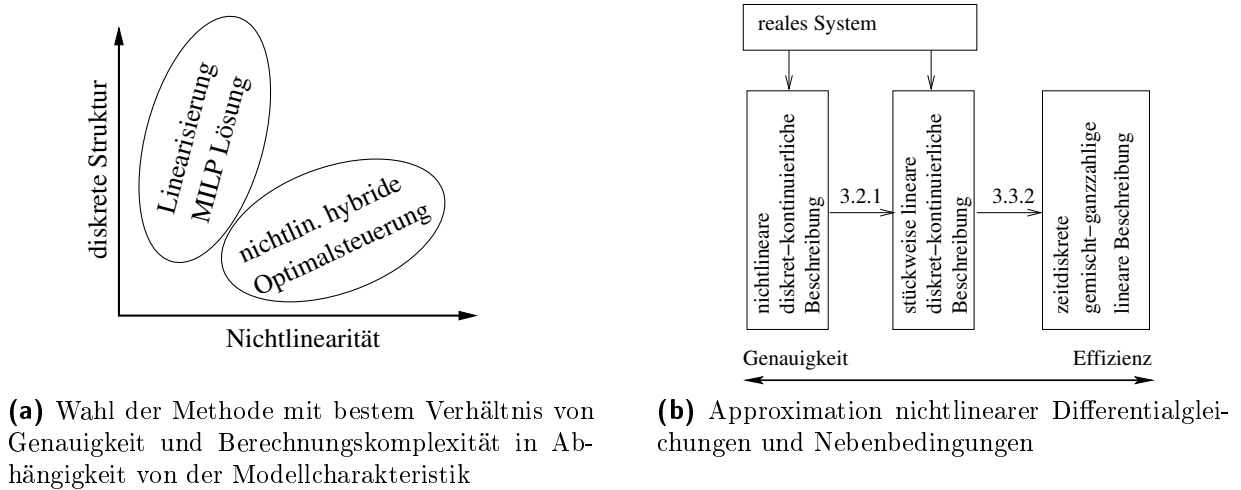


Abbildung 3.2: Entscheidungskriterien und Schritte der Approximation.

3.2.1 Hybridisierungskonzepte nichtlinearer Systemanteile

Im Rahmen dieser Arbeit werden speziell Approximationen und zugehörige Verfahren betrachtet, die auf einer Hybridisierung der nichtlinearen Modellanteile basieren. Hybridisierung meint zunächst die Partition des Steuer-Zustands-Raumes für einen jeweils festen diskreten Parameter q und anschließend das Festlegen darauf gültiger vereinfachter Systemdynamiken.

In [ADG07] wird dieses Vorgehen systematisch beschrieben und zur approximativen Analyse nichtlinearer Systeme vorgeschlagen. Die Hauptidee der Hybridisierung ist demnach die Approximation der komplexen kontinuierlichen Systemdynamik durch eine Auswahl einfacherer kontinuierlicher Dynamiken. „Einfach“ meint Typen der Dynamik, welche leichter und effizienter analysiert werden können, beispielsweise die Approximation einer nichtlinearen Funktion durch eine stückweise affine Dynamik. Dadurch entsteht ein hybrides System, dessen Analyse Informationen über das Originalsystem liefern kann. Neben der Bedeutung in der Systemanalyse ist die Hybridisierung auch für approximative Reglerentwürfe relevant.

Beispielhaft wird in [ADG07] unter anderem eine stückweise affine Approximation auf Simplex einer nichtlinearen, zweimal stetig differenzierbaren Dynamik $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ dargestellt. Ein Simplex M_l ist durch $(n_x + 1)$ affin unabhängige Punkte in $\Omega \subset \mathbb{R}^{n_x}$ festgelegt. Simplexes werden dort basierend auf einer äquidistanten Gittereinteilung (Maschenweite h) von Ω bestimmt. Die affine Approximation wird lokal so gewählt, dass sie an den Eckpunkten des Simplex interpoliert. Dazu wird in [ADG07] die Fehlerabschätzung

$$\varepsilon(h) = \sup_{M_l} \sup_{\mathbf{x} \in M_l} \|\mathbf{f}(\mathbf{x}) - \mathbf{A}_l \mathbf{x} + \mathbf{b}_l\| \leq \frac{K n_x^2}{2(n_x + 1)^2}, \quad (3.3)$$

bewiesen, wobei $K = \max_{i \in \{1, \dots, n_x\}} \sup_{\mathbf{x} \in \Omega} \sum_{p_1=1}^{n_x} \sum_{p_2=1}^{n_x} \left| \frac{\partial^2 f^i(\mathbf{x})}{\partial x_{p_1} \partial x_{p_2}} \right|$.

Eine günstigere Approximation aus Sicht des Verhältnisses von Stützstellen und erzielter Approximationsgenauigkeit wird in [AIS06] vorgestellt. Die Partition des Steuer-Zustands-Raumes wird dort auf Basis eines Gitters im Zielbereich $\mathbf{f}(\mathbb{X} \times \mathbb{U})$ gebildet und erstellt

somit die Partition unter Berücksichtigung der tatsächlichen Werte von $\mathbf{f}(\mathbf{x}, \mathbf{u})$. Der resultierende Approximationsfehler ist dann unmittelbar durch die Maschenweite des Gitters im Zielbereich gegeben.

Im Rahmen der vorliegenden Arbeit werden zwei Ansätze der Transformation nichtlinearer Systemdynamik in stückweise affine Beschreibungen entwickelt:

1) Gemäß einer Heuristik wird eine Auswahl von Punkten $(\mathbf{x}_j^*, \mathbf{u}_j^*)$ im Steuer-Zustands-Raum eines festen diskreten Wertes q festgelegt, in dessen Umgebung M_j die Linearisierung

$$\mathbf{f}_q(\mathbf{x}, \mathbf{u}) \approx \hat{\mathbf{f}}_q = \left. \frac{\partial \mathbf{f}_q}{\partial \mathbf{x}} \right|_{\mathbf{x}_j^*, \mathbf{u}_j^*} (\mathbf{x} - \mathbf{x}_j^*) + \left. \frac{\partial \mathbf{f}_q}{\partial \mathbf{u}} \right|_{\mathbf{x}_j^*, \mathbf{u}_j^*} (\mathbf{u} - \mathbf{u}_j^*) + \mathbf{f}_q(\mathbf{x}_j^*, \mathbf{u}_j^*)$$

das Approximationsmodell markiert. Der gültige Bereich M_j der lokalen, affinen Approximationen wird durch Voronoi-Zellen festgelegt. Diese PWA-Approximation ist im Allgemeinen nicht stetig. Mit Hilfe des Satzes von Taylor lässt sich der Fehler analog zu Gleichung (3.3) abschätzen.

2) Die PWA-Approximation wird durch Optimierung einer festen Anzahl an Simplizes $\mathbb{M}_q = \{M_1, M_2, \dots, M_L\}$ bestimmt. Ausgangspunkt ist dabei eine Überdeckung des Steuer-Zustands-Raumes mit Elementen aus \mathbb{M}_q , wobei für je zwei Elemente aus \mathbb{M}_q gilt: $(M_{l_1} \cap M_{l_2}) \subset (\partial M_{l_1} \cap \partial M_{l_2})$.

Durch Lösen eines nichtlinearen Optimierungsproblems

$$\min_{\{(\mathbf{x}_j, \mathbf{u}_j, \mathbf{z}_j) \mid j=1, \dots, n_M\}} \sum_{l=1}^L \int_{M_l} \|f(\boldsymbol{\xi}, \boldsymbol{\nu}) - (\mathbf{A}_l \boldsymbol{\xi} + \mathbf{B}_l \boldsymbol{\nu} + \mathbf{b}_l)\|_2^2 d(\boldsymbol{\xi}, \boldsymbol{\nu}) \quad (3.4)$$

lassen sich die Eckpunkte $(\mathbf{x}_j, \mathbf{u}_j)$ und ein zugehöriger Wert \mathbf{z}_j derart bestimmen, dass das Integral über den quadratischen Fehler der PWA-Approximation minimiert wird.

Dabei hängen die Einträge der Matrizen \mathbf{A}_l , \mathbf{B}_l und \mathbf{b}_l jeweils selbst von den $n_x + n_u + 1$ Ecken des Simplex M_l ab, so dass folgende Nebenbedingungen des Optimierungsproblems eingehalten werden:

- An den Eckpunkten $(\mathbf{x}_j^l, \mathbf{u}_j^l)$ ($j = 1, \dots, n_x + n_u + 1$) gilt: $\mathbf{z}(\mathbf{x}_j^l, \mathbf{u}_j^l) = \mathbf{A}_l \mathbf{x}_j^l + \mathbf{B}_l \mathbf{u}_j^l + \mathbf{b}_l$.
- Die Menge aller Simplizes $\mathbb{M}_q = \{M_1, M_2, \dots, M_L\}$ überdeckt den zu einem festen q gehörigen Steuer-Zustands-Raum.
- Alle \mathbf{x}_j^l und \mathbf{u}_j^l erfüllen die Bedingungen (3.2b) und (3.2c).

Die Optimalsteuerung auf PWA-Systemen reagiert sehr sensitiv auf eine Erhöhung der Zahl an Polygonen L . Die vorgeschlagene Minimierung, welche im Vorfeld der eigentlichen Lösung des approximativen Optimalsteuerungsproblems durchgeführt wird, erlaubt es, eine im Sinne der Gleichung (3.4) optimale Abwägung von Approximationsfehler und Zahl der Stützstellen zu finden (vgl. auch Abb. 3.3).

Die Berechnung der Punkte $(\mathbf{x}_j, \mathbf{u}_j, \mathbf{z}_j)$ lässt sich durch den Anwender hauptsächlich beeinflussen über die Festlegung der jeweils einen Simplex aufspannenden Eckpunkte. Weiterhin können einzelne Werte für $(\mathbf{x}_j, \mathbf{u}_j, \mathbf{z}_j)$ als konstant festgesetzt werden oder zur Optimierung in (3.4) Norm oder Gewichtungen angepasst werden.

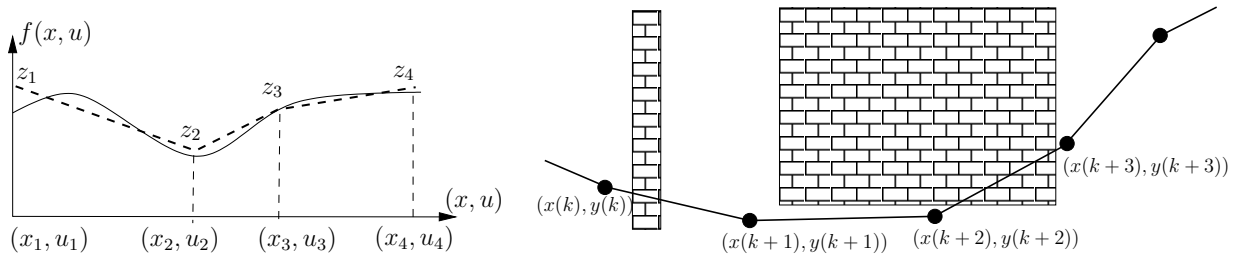


Abbildung 3.3: Optimale Stützstellen einer Approximation (links) und Informationsverlust durch grobe Diskretisierung (rechts): Trotz zulässiger Wegpunkte (x, y) verläuft die Trajektorie durch Teile von Hindernissen.

Lineare Formulierung von (Un-)Gleichungsnebenbedingungen: Die Bedingungen (3.2c) und (3.2b) weisen jedem diskreten Zustand q eine Teilmenge des Steuer-Zustands-Raumes $\mathbb{X} \times \mathbb{U}$ zu. Um ein vollständig lineares Systemmodell zu erhalten, werden auch diese Restriktionen des diskret-kontinuierlichen Optimalsteuerungsproblems linear approximiert. Wählt man eine Approximation in PWA-Beschreibung, so ist es zweckmäßig, die zu beschreibende Menge in Einklang mit den Simplexes der Dynamik-Approximation zu wählen. Dabei muss häufig anhand der Bedeutung im System entschieden werden, ob ein Detail konservativ, progressiv oder nach minimalem Fehler zu approximieren ist.

Abschließend sei noch erwähnt, dass für praktische Anwendungen ein PWA-Modell auch durch Identifikationsmethoden (zum Beispiel [FTMLM01]) gewonnen werden kann, wenn Details des Systemmodells unbekannt sind.

3.2.2 Zeitdiskretisierung und Euler-Approximation

In Abschnitt 2.3 wurde einleitend der Übergang von einfachen, zeitkontinuierlichen auf zeitdiskrete lineare Systeme beschrieben. Ein weiterer Zugang zu zeitdiskreten Systemen lässt sich über die einfache Euler-Diskretisierung herstellen,

$$\frac{\mathbf{x}(t_{k+1}) - \mathbf{x}(t_k)}{\Delta_t} \approx \dot{\mathbf{x}}(t_k) = \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k))$$

und schließlich

$$\mathbf{x}(t_{k+1}) \approx \Delta_t \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) + \mathbf{x}(t_k) .$$

Der Übergang von der Funktion $\mathbf{x}(t)$ zur endlichen Menge an Stützstellen $\mathbf{x}(t_k)$ zieht einen gewissen Informationsverlust nach sich. So ist grundsätzlich zu prüfen, ob dieser durch zusätzliche Nebenbedingungen ausgeglichen werden muss. Anschaulich zeigt Abb. 3.3 eine solche Problematik.

Die Zahl an Zeitschritten, in die das Intervall $[t_0, t_f]$ eingeteilt wird, ist im Allgemeinen deutlich größer zu wählen als die Zahl an Phasen im Optimalsteuerungsproblem (s. Definition 9), um eine vertretbare Approximationsgüte zu erlangen. Aus Sicht der Automatenmodelle entstehen auf festem Zeitgitter nun getaktete Automaten. Da sich mit wachsender Zahl an Zeitschritten n_c direkt die Zahl an möglichen diskreten Übergängen erhöht, entsteht ein größerer Anteil diskreter Struktur im Gesamtmodell. Diese steht der Verbesserung numerischer Eigenschaften entgegen, die durch die Linearisierung beabsichtigt wird.

Auf zeitdiskreten Modellen zu arbeiten, erweist sich als zielführend im Hinblick auf modellprädiktive Methoden, die in Abschnitt 3.3 diskutiert werden. Andererseits beschränkt sich auch das Feld der effizient zu beantwortenden Fragestellungen. So kann die Frage nach zeitoptimaler Regelung prinzipiell beantwortet werden, allerdings führt das starre Zeitgitter dabei zu einer aus Optimierungssicht ungünstigen Problemstruktur.

Stückweise lineare zeitdiskrete Systeme: In der Literatur werden stückweise lineare zeitdiskrete Modelle überwiegend als PWA- und MLD-Systeme betrachtet, die im Abschnitt 2.3 vorgestellt wurden. Im Kontext der Steuerung kooperativer Fahrzeugsysteme wurden stückweise lineare zeitdiskrete Modelle zu spezifischen Problemstellungen vorgeschlagen, z.B. in [RH02, ED04]. Eine Auswahl der dort vorgestellten Ansätze werden in der vorliegenden Arbeit in Systematik, in Modellierungsdetails und im Anwendungsspektrum vertieft, verallgemeinert und weiterentwickelt.

Das breite Interesse an dieser Modellierung, die zunächst sehr approximativ und einschränkend erscheint, ist im Lauf der Jahre mit der Entwicklung effizienter MILP-Löser gewachsen. Die Vorzüge von MILP-Modellen bestehen in der effizienten, stets globalen Optimierung bei einer sehr geringen Abhängigkeit von Startschätzungen. Darauf basiert die Leistungsfähigkeit zahlreicher Ansätze zur Regelung diskret-kontinuierlicher Systeme, z.B. [BM99, SE02, MFK06, Kva08, Chr06] .

3.2.3 Vorschläge zum Einsatz von approximativen Modellen

Die Motivation, Modelle zu approximieren, ist es die Effizienz der Lösungsfindung zu steigern. Für die in Kapitel 5 vorgestellten Probleme mit teils \mathcal{NP} -schwerem Charakter stellt der Rückzug auf Approximationsmodelle dabei oft sogar eine Notwendigkeit dar, um überhaupt mit derzeit verfügbaren Computern in akzeptablen Zeiten näherungsweise Lösungen berechnen zu können. Dabei bedarf es einer problemspezifischen Abwägung von nötiger Modellgenauigkeit und vertretbarem Rechenaufwand. Dieser Abwägungsprozess wird durch den modularen Aufbau und skalierbare Teilmodelle in den vorgestellten Konzepten unterstützt.

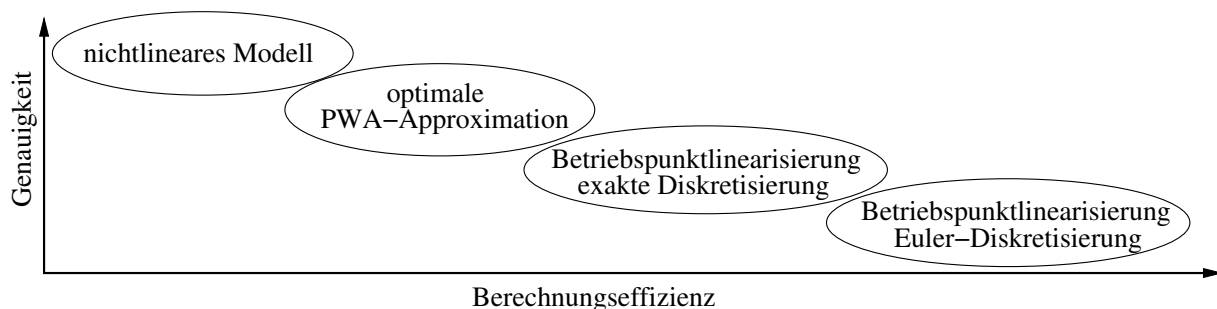


Abbildung 3.4: Abwägung von Genauigkeit und Berechnungseffizienz in der Modellapproximation.

Mit approximativen Modellen ist es möglich, Schwierigkeiten der nichtlinearen diskret-kontinuierlichen Optimalsteuerung zu umgehen und dem Anwender hilfreiche Informationen bereit zu stellen.

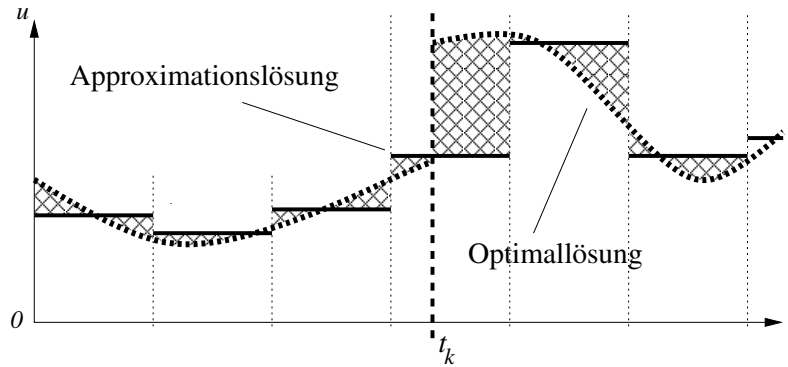


Abbildung 3.5: Abweichung der approximativen von der optimalen Steuersequenz.

Der Kollokationsansatz in Abschnitt 3.1.3 (vgl. auch [Glo05]) erlaubt es, hybride Bewegungsdynamikmodelle mit hoher Nichtlinearität zu lösen. Die Performanz der numerischen Gesamtmethode hängt maßgeblich von Startschätzungen und Schranken für die Teilprobleme der inneren Schleife ab. Im Allgemeinen kann das SQP-Verfahren dort allerdings nur lokale Optimalität garantieren. Die Lösung zeitdiskreter linearer Modelle hingegen erweist sich ohne Vorgabe von Startwerten als effizienter (vgl. Ergebnisse in 5.1 oder auch in [BFV06]) und liefert ein globales Optimum. Sofern eine PWA-Approximation mit zufriedenstellender Genauigkeit bei moderater Zahl an Regionen (Simplizes) erreicht werden kann, bietet sich die Methode somit zur Generierung von Startschätzungen an und kann damit neben der Effizienz des Kollokationsansatzes auch die Globalitätseigenschaft berechneter Optima deutlich verbessern. Insbesondere bei Nichtkonvexitäten im Modell, etwa durch Hindernisbeschränkungen, wird dieser Vorteil offensichtlich.

Optimalitätsverlust bei Optimierung auf approximativen Modellen: Ausgehend vom realen kooperativen Mehrfahrzeugsystem fassen wir die auftretenden Modellierungs- und Approximationsfehler und Optimalitätsverluste zusammen, welche eine auf dem Ersatzmodell berechnete Steuersequenz gegenüber der realen Optimallösung besitzt. Das Maß für den auftretenden Optimalitätsverlust ergibt sich bei gegebenen Anfangsbedingungen als Wert der Zielfunktion (3.1) mit den approximativ berechneten Steuerungen und den sich daraus ergebenden Zustandsverläufen, verglichen mit dem globalen Optimum von (3.1).

Die Fehlerquellen lassen sich wie folgt charakterisieren (vgl. Abb. 3.5):

1. Festlegung diskreter Schaltungen auf ein festes Zeitgitter:
Die zulässige Menge der diskreten Zustände \mathbb{Q} und deren Abfolgen lassen sich exakt linear beschreiben. Durch die Festlegung eines Zeitgitters werden jedoch die Schaltzeitpunkte t_k auf das Zeitgitter gezwungen. Ist die Berechnung aller \mathbf{x} und \mathbf{u} in der Umgebung des Zeitpunktes t_k exakt, ergibt sich lokal eine Abweichung des Schaltzeitpunktes von maximal $\frac{\Delta t}{2}$ Zeiteinheiten.
2. PWA-Approximation von Dynamik und (Un-)Gleichungsnebenbedingungen:
Eine Abschätzung des auftretenden Fehlers ist auf Seite 36 diskutiert. Durch eine iterative Verfeinerung der PWA-Approximation im Bereich einer berechneten Lösung kann der Fehler reduziert werden. Ein Ansatz dazu wurde beispielsweise auch in [ED04] vorgestellt.

3. Zeitdiskretisierung kontinuierlicher Systementwicklung:
Eine Änderung des Systemverhaltens zieht den Übergang zu stückweise konstanten Steuerungen \mathbf{u} nach sich. Bei stückweise konstanter Steuerung geschieht die Diskretisierung linearer Modelle frei von Approximationsfehlern (s. Abschnitt 2.3).
4. Linearisierung der Zielfunktion:
Die Approximation der Zielfunktion muss idealerweise so konstruiert sein, dass diese im Minimum eine möglichst hohe Approximationsgenauigkeit erreicht. Kann die Lage dieses Minimums zum Zeitpunkt der Approximation nicht hinreichend genau abgeschätzt werden, so ist eine adaptive Anpassung auf Basis einer berechneten Lösung möglich.

Um die genannten Approximationsfehler zu verringern, empfiehlt sich eine sukzessive Modellverbesserung (vgl. Abschnitt 3.4) durch neue Zeitpunkte, und eine verbesserte PWA-Approximation im Bereich der bereits gefundenen Lösung. Im Abschnitt 3.3 werden wir stabilisierende Eigenschaften modellprädiktiver Reglerstrukturen benennen. Diese führen auch zu einer teilweisen Kompensation der Approximationsfehler.

Neben den untersuchten Ansätzen sind in der Literatur eine Vielzahl weiterer Methoden zur Approximation nichtlinearer dynamischer Systeme bekannt. So wird das Problem der Approximation hybrider dynamischer Systeme zusammen mit einer Fehlerabschätzung in [GJP08] systematisch diskutiert. Konkrete Ansätze basieren beispielsweise auf geometrischen Betrachtungen, auf sukzessiver Betriebspunkt-Linearisierung [SE02] oder auf maßtheoretischen Betrachtungen [Rub86, MFK06].

Die Lösbarkeit zunehmend realitätsnaher Problemstellungen stellt den mittelfristigen Einsatz der vorgestellten Verfahren zur Aufgaben- und Trajektorienplanung in vielen realen Systemen in Aussicht. Eine Abwägung von Berechnungseffizienz und Modellgenauigkeit ist möglich. Das folgende Kapitel behandelt dazu weiterführende Fragen der Anwendung von MILP-Modellen im Rahmen modellprädiktiver Regler.

3.3 Modellprädiktive Regelung (MPC) diskret-kontinuierlicher linearer Modelle

Dieser Abschnitt bereitet eine neuartige Anwendung von modellprädiktiven Regelungskonzepten auf Kernprobleme kooperativer Mobilität in 5.3 vor.

3.3.1 Prinzip der modellprädiktiven Regelung

Die bisher vorgestellten Methoden der Optimalsteuerung berücksichtigen noch keine Zustandsrückführung. In der Anwendung der optimalen Steuerungen auf reale Systeme ist es allerdings von entscheidender Bedeutung, Unbestimmtheiten und Störungen in der tatsächlichen zeitlichen Entwicklung des System möglichst gut zu kompensieren. Durch eine Rückführung des tatsächlichen Systemzustandes in die Berechnung der Optimalsteuerung lässt sich die Methodik der *modellprädiktiven Regelung* oder auch „Model (Based) Predictive Control“ (MPC) (bzw. „Receding horizon control“ (RHC)) beschreiben. Dabei wird mit Verfahren der Echtzeitoptimierung über einen endlichen Zeithorizont $[t_k, t_k + H]$ ein Optimalsteuerungsproblem gelöst und nur die Anfangssequenz der berechneten Steuerfolge an

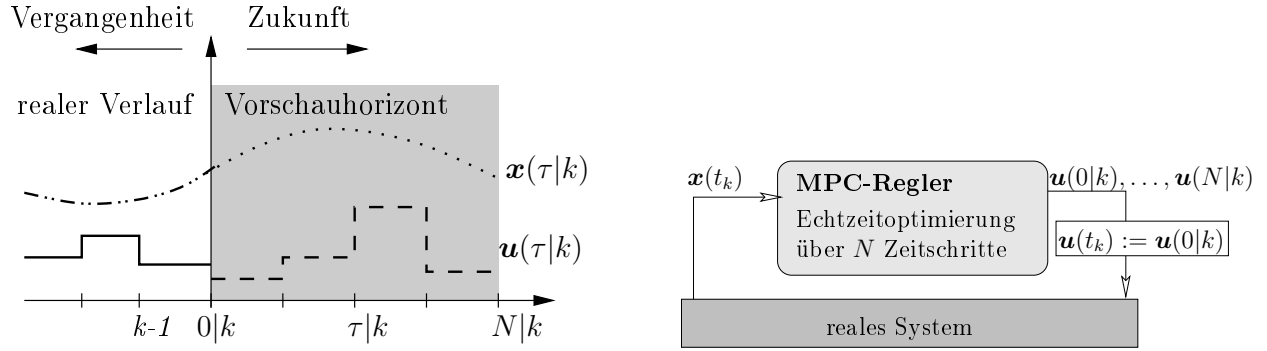


Abbildung 3.6: Grundidee der MPC. Ziel ist es, optimale Steuerungen u_k bis u_{k+N} zu berechnen, so dass über den Horizont von N Zeiteinheiten eine Zielfunktion minimiert wird. Durch die Zustandsrückführung wird aus der Steuerung eine Regelung.

das System übergeben. Im nächsten Schritt wird dann der aktuelle, reale Zustand x_k als Anfangszustand betrachtet und die Echtzeitoptimierung mit verschobenem Zeithorizont fortgesetzt (siehe auch Abb. 3.6). Für den zeitdiskreten Fall lautet der Basisalgorithmus der MPC mit der Notation $x(\tau|k) := x(k + \tau)$ wie folgt:

Algorithmus 1: Zeitdiskreter Basisalgorithmus der modellprädiktiven Regelung

- 1: **while** System läuft **do**
 - 2: $x(0|k) \leftarrow x(t_k)$ (gemessener) Systemzustand zur aktuellen Zeit t_k ;
 - 3: löse Optimalsteuerungsproblem für $t \in [t_{0|k}, t_{N+1|k}]$ mit Startwert $x(0|k)$;
 - 4: führe $u(0|k)$ der optimalen Steuersequenz $(u(\tau|k))_{\tau=0, \dots, N}$ aus;
 - 5: warte bis zum Zeitpunkt t_{k+1} ;
 - 6: $k \leftarrow k + 1$;
 - 7: **end while**
-

Die Optimalität der MPC wird lediglich durch die Wahl des Zeithorizontes und durch Modellapproximationen beeinträchtigt. Die sich daraus ergebende große Flexibilität verleiht der MPC ein weites Spektrum potentieller Anwendungen. MPC eignet sich für beschränkte Mehrgrößen-Regelungsprobleme, bei denen die Offline-Berechnung eines allgemeinen Regelgesetzes sehr schwierig oder sogar unmöglich ist. MPC-Regler werden bevorzugt in verfahrenstechnischen Prozessen genutzt, in welchen die relevanten Systemdynamiken langsam genug sind, um eine Optimierung in jedem Abtastschritt durchführen zu können. Einer der großen Vorteile von MPC ist die Fähigkeit, Zustands- und Steuerbeschränkungen zu behandeln, was die Methode für Anwendungen im Bereich der kooperativen Mobilität sehr interessant macht.

Ein weiterer Vorzug modellprädiktiver Regelung ergibt sich durch die Möglichkeit im dem zu Grunde liegenden Optimierungsproblem das Modell einer sich dynamisch ändernden Umgebung aufzunehmen. In Abschnitt 5.3 wird diese Möglichkeit beispielsweise zur Beschreibung eines Gegenspieler im Roboterfußball angewendet.

In der Literatur werden MPC-Ansätze für allgemeine, rein kontinuierliche, nichtlineare Systeme ausführlich diskutiert. Dabei beschränken sich die meisten Untersuchungen auf stückweise konstante Steuerfunktionen mit äquidistantem Zeitgitter, derer sich Abschnitt 3.3.3 für den linearen Spezialfall widmet.

Untersuchungen allgemeiner, nichtlinearer MPC-Probleme umfassen die Lösung mit direkten und indirekten Verfahren (vgl. Abschnitt 3.1.3), aber auch Suboptimalitätsabschätzungen (z.B. [GP09]). Einen Überblick zur nichtlinearen MPC bietet [FBA06]. Die Grenzen der MPC lassen sich durch das Verhältnis der System-Zeitskalen und der Rechenzeit zur Lösung des Optimalsteuerungsproblems ausloten.

Im Kontext kooperierender Fahrzeuge wurden diskret-kontinuierliche MPC-Methoden bisher nur in hierarchisch zerlegten [BF06] und vollständig verteilten bzw. dezentralen Regelungsstrukturen vorgestellt. Einen allgemeinen Überblick zur verteilten MPC in Multi-Agenten-Systemen bietet der Artikel [NDH04]. Die Grundidee verteilter Verfahren ist das gemeinsame Optimieren einer Zielfunktion, gegebenenfalls in einer hierarchischen Systemmodellierung. Die Kommunikation der Fahrzeuge untereinander ist dabei von großer Bedeutung, wie dies beispielsweise die Arbeiten [DS09b, DS09a, KBF⁺08] aufzeigen.

Die optimale Regelung der kooperativen Aufgabenverteilung mit Hilfe der MPC unter Berücksichtigung der Fahrzeugdynamik wurde bisher in der Literatur noch kaum untersucht. Eine Motivation der vorgestellten Arbeiten ist das Ausloten der Machbarkeit, wozu der Abschnitt 3.3.3 den diskret-kontinuierlichen linearen Fall genauer betrachtet.

Zunächst werden jedoch noch einmal die Struktur und Lösungsmethodik gemischt-ganzzahliger linearer Optimierungsprobleme diskutiert, die als Kern der angestrebten Echtzeitoptimierung besondere Bedeutung besitzt.

3.3.2 Gemischt-ganzzahlige lineare Optimierung (MILP)

Aufgabe der gemischt-ganzzahligen Optimierung (MILP) ist es, Werte \mathbf{x}_c und \mathbf{x}_l zu bestimmen, so dass die lineare Zielfunktion

$$\min_{\mathbf{x}_c, \mathbf{x}_l} \mathbf{c}_c^T \mathbf{x}_c + \mathbf{c}_l^T \mathbf{x}_l$$

unter Einhaltung der linearen Nebenbedingungen

$$\mathbf{A}_c^T \mathbf{x}_c + \mathbf{A}_l^T \mathbf{x}_l \leq \mathbf{b}$$

mit $\mathbf{x}_c \in \mathbb{R}^{\tilde{n}_c}$ und $\mathbf{x}_l \in \mathbb{Z}^{n_l}$ minimiert wird. Im Rahmen dieser Arbeit wird der Fall mit $\mathbf{x}_l \in \{0, 1\}^{n_l}$ und Vektoren $\mathbf{c}_c \in \mathbb{R}^{\tilde{n}_c}$, $\mathbf{c}_l \in \mathbb{R}^{n_l}$, $\mathbf{b} \in \mathbb{R}^{n_b}$ sowie Matrizen $\mathbf{A}_c \in \mathbb{R}^{n_b \times \tilde{n}_c}$ und $\mathbf{A}_l \in \mathbb{R}^{n_b \times n_l}$ betrachtet.

Von Natur aus ist das Problem – aufgrund der Bedingung $\mathbf{x}_l \in \{0, 1\}^{n_l}$ – nicht konvex und \mathcal{NP} -vollständig. Betrachtet man hingegen das relaxierte Problem mit $\mathbf{x}_l \in [0, 1]^{n_l}$, so zeigt sich ein rein kontinuierliches lineares Optimierungsproblem (LP), für das hocheffiziente Lösungsalgorithmen existieren.

Auch zur Lösung von MILP sind seit einigen Jahren sehr effizient arbeitende Programme verfügbar, was das Interesse in Wissenschaft und Industrie an der Modellierung und Regelung diskret-kontinuierlicher Systeme sehr erhöht hat. Insbesondere die bereits vorgestellten MLD- und PWA-Systeme finden dabei Beachtung.

Moderne Solver bedienen sich hauptsächlich rekursiver Baum-Such-Verfahren, aber auch Lösungen mit genetischen Ansätzen und „simulated annealing“-Verfahren werden immer wieder präsentiert. Einige kommerzielle MILP-Löser wie CPLEX [ILO07] oder Xpress-MP [SMY⁺] bieten mittlerweile Routinen, die den Benutzer automatisiert dabei unterstützen, für eine gegebene MILP-Struktur eine effiziente Lösungsstrategie mit zugehörigen Parametern zu wählen. Ferner existieren für Kernprobleme des Operations Research, wie etwa

das „Rucksack-Problem“ oder das „Handlungsreisenden-Problem“ (vgl. Abschnitt 4.4.2), hocheffiziente spezielle Algorithmen.

Wird das MILP durch einen diskreten (binärwertigen) Anteil dominiert, so zeigen sich kombinierte Ansätze der Optimierung und der Logikprogrammierung häufig als eine sehr effiziente Variante (z. B. [BG06]).

MILP als Basismethode der modellprädiktiven Regelung in hybriden dynamischen Systemen: MILP erlaubt es, diskrete Modellanteile und eine einfache kontinuierliche Dynamik simultan zu berücksichtigen. Zur Lösung stehen effiziente Algorithmen bereit, die relativ komfortabel in der Anwendung sind: es brauchen keine Ableitungen bereit gestellt zu werden, die Abhängigkeit von Startschätzungen ist sehr gering, und es existieren hocheffiziente Löser, welche ein globales Optimum liefern.

Dem entgegen steht das grundsätzliche Problem der diskret-kontinuierlichen Optimierung, dass die Berechnungseffizienz äußerst sensitiv auf die Art der Problemformulierung reagiert. Die Bücher [Kal02] und [Hoo07] vertiefen die angesprochene Thematik und widmen sich unter anderem systematisch der Problematik guter Modellierung.

Die prinzipielle Eignung MILP-basierter Methoden zur Aufgabenzuweisung sowie zur Trajektorienplanung lässt sich auch anhand verschiedenartiger exemplarischer Problemlösungen auf MILP-Basis in der Literatur ablesen (z.B. [MFK06, SFH06, ED07]).

3.3.3 Algorithmen und Eigenschaften MILP-basierter MPC-Methoden

MILP kann zu komplexen Planungsaufgaben der Kooperation global-optimale Lösungen finden. Auch Unbestimmtheiten lassen sich dabei über freie Variablen berücksichtigen. Die Notwendigkeit der MPC, in Echtzeit eine (Sub-)Optimallösung eines gegebenenfalls approximativen diskret-kontinuierlichen Optimalsteuerungsproblems zu generieren, setzt hohe Anforderungen an den MILP-Löser. Da die betrachteten Probleme \mathcal{NP} -schwer sind, kann die Berechnung in Einzelfällen sehr lange dauern. Mit guter, problemspezifischer Modellierung lässt sich die Rechenzeit meist in einem vertretbaren Rahmen halten, aber auch hier sind Grenzen gesetzt.

Neben dem Einbringen problemspezifischen Wissens wird im Allgemeinen versucht, mit approximativen „cost-to-go“-Funktionen und multiparametrischen Vorabberechnungen die Echtzeitfähigkeit zu erreichen.

Explizite, multiparametrische Berechnung des MPC-Reglers: Die Echtzeitoptimierung der MPC ist auf rechenschwachen Plattformen (z.B. auf mobilen Robotern) oft kaum praktikabel; insbesondere bei Verwendung hochdynamischer, hybrider Systemmodellen. Ein Ansatz, die Echtzeitfähigkeit ohne Optimalitätsverlust zu gewährleisten, besteht in der multiparametrischen Vorabberechnung des kompletten MPC-Gesetzes für alle möglichen Systemsituationen \mathbf{x}_0 , vgl. Abb. 3.7. Die Grundlage dafür bildet der folgende Satz:

Satz 1. ([Bor03], Kapitel 8.4)

Gegeben sei das PWA-System (vgl. Definition 7) auf der polyedrischen Partition $\{\mathbf{P}_i\}_{i=1}^s$, so dass gelte

$$\forall i = 1, \dots, s : \begin{pmatrix} \mathbf{x}(\tau|k) \\ \mathbf{u}(\tau|k) \end{pmatrix} \in \mathbf{P}_i \Rightarrow \mathbf{x}(\tau+1|k) = \mathbf{A}^i \mathbf{x}(\tau|k) + \mathbf{B}^i \mathbf{u}(\tau|k) + \mathbf{f}^i. \quad (3.5)$$

Des weiteren sei eine Endzustandsmenge \mathbb{X}_f gegeben. Es bezeichne $\|\mathbf{x}\|_M$ eine der Normen $\|\mathbf{M}\mathbf{x}\|_1$, $\|\mathbf{M}\mathbf{x}\|_\infty$ oder $\mathbf{x}^T \mathbf{M} \mathbf{x}$ mit positiv definiten Matrix \mathbf{M} , dann gilt:

Die Lösung des Optimalsteuerungsproblems

$$\min_{(\mathbf{u}(\tau|k))_{\tau=0,\dots,N}} \|\mathbf{x}(N+1|k)\|_{\mathbf{P}} + \sum_{\tau=0}^N \|\mathbf{x}(\tau|k)\|_{\mathbf{Q}} + \|\mathbf{u}(\tau|k)\|_{\mathbf{R}} \quad (3.6)$$

(mit anwendungsspezifisch gewählten Matrizen \mathbf{P} , \mathbf{Q} und \mathbf{R}) unter den Nebenbedingungen (3.5) und $\mathbf{x}(N+1|k) \in \mathbb{X}_f$ ist eine stückweise affine Funktion der Form

$$\mathbf{u}(\tau|k) = \mathbf{F}_\tau^i \mathbf{x}(\tau|k) + \mathbf{g}_\tau^i, \text{ falls } \mathbf{x}(\tau|k) \in \mathcal{C}_\tau^i,$$

wobei die Gebiete \mathcal{C}_τ^i ($\forall i = 1, \dots, s_k : \mathcal{C}_\tau^i \neq \emptyset$) eine Partition der Menge aller im Schritt τ zulässigen Zustände bilden.

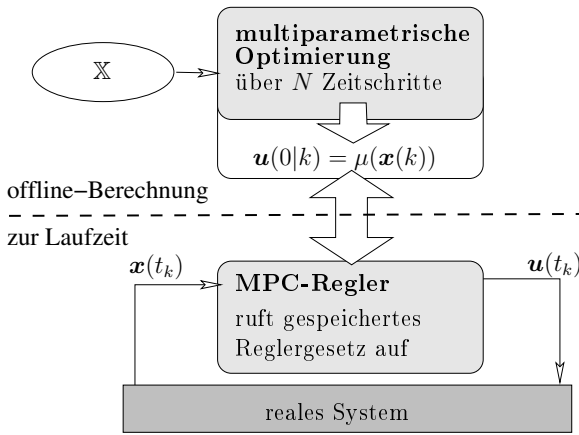


Abbildung 3.7: Konzept der expliziten MPC-Berechnung.

Das MPC-Gesetz kann also mit einer endlichen Zahl an Parametern für alle Systemsituationen \mathbf{x}_0 beschrieben werden. Ziel ist es nun, diese Parameter vorab zu berechnen und dann zur online-Anwendung als look-up zur Verfügung zu stellen. Diese Berechnung leistet beispielsweise die MULTIPARAMETRIC TOOLBOX (MPT) für das Programmpaket MATLAB[Mat]. Die zugehörigen Algorithmen werden in [Chr06] und [Kva08] erläutert. [Hod07] stellt darüber hinaus Ansätze vor, welche die Zahl der Partitionen dabei gering halten können.

Robustheit: Die Robustheit der MPC bei hybriden linearen zeitdiskreten dynamischen Systemen wird meist durch eine Min-Max-Technik sichergestellt: Minimiere die maximalen Kosten, die nötig sind, um die schlimmste Störung auszugleichen. In [TOBD06] wird folgendes konstruktives Vorgehen vorgestellt, mit dem im Reglerdesign für PWA-Systeme Robustheitsanforderungen berücksichtigt werden:

Betrachtet wird dazu eine polyedrische Zielregion (oder auch ein Gleichgewichtszustand) $\mathbf{P}_\tau \subset \mathbb{R}^{n_x} \times \{0, 1\}^{n_t}$. Damit kann $\mathbf{P}_{\tau-1}$ als diejenige Region berechnet werden, aus der das

System trotz aller Störungen $\mathbf{d} \in \mathbf{D}$ in \mathbf{P}_τ überführt werden kann.

$$\mathbf{P}_{\tau-1} := \left\{ \mathbf{x}(\tau-1|k) \mid \exists i \exists \mathbf{u}(\tau-1|k) \in \mathbb{U}, \begin{pmatrix} \mathbf{x}(\tau-1|k) \\ \mathbf{u}(\tau-1|k) \end{pmatrix} \in \mathcal{X}_i; \forall \mathbf{d}^i(\tau-1|k) \in \mathbf{D} : \right. \\ \left. \mathbf{A}^i \mathbf{x}(\tau-1|k) + \mathbf{B}^i \mathbf{u}(\tau-1|k) + \mathbf{C}^i \mathbf{d}^i(\tau-1|k) + \mathbf{f}^i \in \mathbf{P}_\tau \right\}$$

Damit lassen sich in Abhängigkeit des Vorschauhorizontes Mengen konstruieren, unter denen der MPC-Regler robust arbeitet. Dies gilt, solange für die Störungen $\mathbf{d} \in \mathbf{D}$ erfüllt ist und sie in der Form $\mathbf{x}(\tau+1|k) = \mathbf{A}^i \mathbf{x}(\tau|k) + \mathbf{B}^i \mathbf{u}(\tau|k) + \mathbf{C}^i \mathbf{d}^i(\tau|k) + \mathbf{f}^i$ in die Systembeschreibung eingehen. Dabei lässt sich der Fehlerterm $\mathbf{C}^i \mathbf{d}^i(\tau|k)$ und die Menge \mathbf{D} aus den Betrachtungen von Seite 39 ableiten.

Man geht in der Rekursion mindestens so weit zurück, wie der geplante Vorschauhorizont gewählt werden soll. Wird damit nicht der volle Zustandsraum $\mathbb{R}^{n_x} \times \{0, 1\}^{n_i}$ erreicht, muss entweder die Approximation verbessert werden (\mathbf{D} wird kleiner) oder die Endzustandsmenge größer gewählt werden.

Der Zeit- und Speicher-Aufwand zur Berechnung mit dieser Rückwärtsrekursion wächst wiederum exponentiell mit der Zahl an Dynamiken i und der Länge des Vorschauhorizontes. Für viele Fälle wird diese Analyse dadurch nicht mehr durchführbar. Die thematische Nähe zu den Fragen der Erreichbarkeitsanalysen (vgl. S. 22) ist – rückwärts in der Zeit – offensichtlich.

Stabilität: Die Attraktivitätseigenschaft eines MPC-Reglers bei MLD-Systemen bzgl. eines Gleichgewichtspunktes wird in [BM99, BBM00] – in Abhängigkeit der Gewichtungen in der Zielfunktion (in ∞ -Norm) – bewiesen. Zur Komplexität der Echtzeioptimierung weist bereits [BBM00] darauf hin, dass seine Stabilitätsaussagen nicht davon abhängen, dass das tatsächliche Minimum in der MPC angenommen wird und somit auch suboptimale Lösungen die Stabilitätseigenschaften garantieren, sofern eine Annäherung an den Gleichgewichtspunkt vom Übergang von τ nach $\tau+1$ stattfindet. Somit kann ein iterativ arbeitender MILP-Löser auch mit einer Zeitbeschränkung verwendet werden, der die beste im Zeitraumen gefundene Lösung zur Verwendung im MPC-Gesetz zurückgibt.

In [LHWB06] werden folgende Bedingungen angegeben, die eine asymptotische Stabilität sehr allgemeiner hybrider, MPC-geregelter Systeme garantieren.

Satz 2. *Gegeben sei ein zeitdiskretes System*

$$\mathbf{x}(k+1) = \tilde{\mathbf{f}}(\mathbf{x}(k), \mathbf{u}(k)) \quad (\mathbf{x}(k) \in \mathbb{X} \subseteq \mathbb{R}^{n_x}, \mathbf{u}(k) \in \mathbb{U} \subseteq \mathbb{R}^{n_u}). \quad (3.7)$$

Die Mengen \mathbb{X} und \mathbb{U} spezifizieren Zustands- und Steuerbeschränkungen und seien kompakte polyedrische Mengen mit $(\mathbf{0}, \mathbf{0}) \in \text{int}(\mathbb{X} \times \mathbb{U})$. Es gelte $\tilde{\mathbf{f}}(\mathbf{0}, \mathbf{0}) = \mathbf{0}$.

$\mathbb{X}_T \subseteq \mathbb{X}$ (mit $\mathbf{0} \in \mathbb{X}_T$) bezeichne eine gewünschte Zielmenge. Die Menge zulässiger Steuersequenzen der Länge N bezüglich \mathbb{X}_T sei $\mathcal{U}_N(\mathbf{x}(0|k)) := \{(\mathbf{u}(\tau|k))_{\tau=1, \dots, N} \in \mathbb{U}^N \mid \tau=1, \dots, N : \mathbf{x}(\tau|k) \in \mathbb{X} \text{ und } \mathbf{x}(N+1|k) \in \mathbb{X}_T\}$.

Gegeben seien weiterhin die Abbildungen $F : \mathbb{R}^{n_x} \rightarrow \mathbb{R}_{\geq 0}$ mit $F(\mathbf{0}) = 0$ und $L : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}_{\geq 0}$ mit $L(\mathbf{0}, \mathbf{0}) = 0$, welche zu gegebenem $\mathbf{x}(k)$ eine MPC-Zielfunktion

$$J(\mathbf{x}(k), \mathbf{u}(k)) := F(\mathbf{x}(N+1|k)) + \sum_{\tau=1}^N L(\mathbf{x}(\tau|k), \mathbf{u}(\tau|k))$$

des Systems (3.7) über die Menge $\mathcal{U}_N(\mathbf{x}(k))$ definiert.

Sei $\mathbb{X}_f(N) := \{\mathbf{x} \in \mathbb{X} \mid \mathcal{U}_N(\mathbf{x}) \neq \emptyset\}$. Sei weiterhin $h : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$ mit $\mathbb{X}_U := \{\mathbf{x} \in \mathbb{X} \mid h(\mathbf{x}) \in \mathbb{U}\}$ eine beliebige Funktion, für die $h(\mathbf{0}) = \mathbf{0}$ gilt.

Es existieren Funktionen $\alpha_1(\cdot), \alpha_2(\cdot) \in \mathcal{K}^2$, eine Nachbarschaft $\mathcal{N} \subseteq \mathbb{X}_f(N)$ um $\mathbf{0}$ und die Funktion $h(\cdot)$ in der Form, dass $\mathbb{X}_T \subseteq \mathbb{X}_U$ (mit $\mathbf{0} \in \text{int}(\mathbb{X}_T)$) eine positiv invariante Menge ¹ des Systems (3.7) in geschlossener Form mit $\mathbf{u}(k) = h(\mathbf{x}(k))$ ist, so dass gilt

$$\forall \mathbf{x} \in \mathbb{X}_f, \forall \mathbf{u} \in \mathbb{U} : \quad L(\mathbf{x}, \mathbf{u}) \geq \alpha_1(\|\mathbf{x}\|) \quad (3.8a)$$

$$\forall \mathbf{x} \in \mathcal{N} : \quad F(\mathbf{x}) \leq \alpha_2(\|\mathbf{x}\|) \quad (3.8b)$$

$$\forall \mathbf{x} \in \mathbb{X}_T : \quad F(\tilde{\mathbf{f}}(\mathbf{x}, h(\mathbf{x}))) - F(\mathbf{x}) + L(\mathbf{x}, h(\mathbf{x})) \leq 0. \quad (3.8c)$$

Dann gilt:

- i) Falls $\mathcal{U}_N(\mathbf{x}(k)) \neq \emptyset$ folgt $\mathcal{U}_N(\tilde{\mathbf{f}}(\mathbf{x}(k), \mathbf{u}^*(k))) \neq \emptyset$ und damit $\mathbb{X}_T \subseteq \mathbb{X}_f(N)$
- ii) Der Ursprung des geschlossenen, MPC-geregelten Systems ist für alle Startwerte in $\mathbf{x}(k) \in \mathbb{X}_f(N)$ asymptotisch stabil nach Lyapunov. ¹
- iii) Sind obige Ungleichungen mit $\alpha_1(s) = a s^\lambda$, $\alpha_2(s) = b s^\lambda$ für Konstanten $a, b, \lambda > 0$ erfüllt, so ist der Ursprung des geschlossenen, MPC-geregelten Systems für alle Startwerte in $\mathbf{x}(k) \in \mathbb{X}_f(N)$ exponentiell stabil.

Bemerkung 11. An die Funktionen $\tilde{\mathbf{f}}(\cdot, \cdot) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ und $h : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$ sind keine weiteren Forderungen gestellt. Insbesondere sind Nichtlinearitäten und Unstetigkeiten zulässig.

Bemerkung 12. Die Stabilitätseigenschaft gilt auch, wenn das Optimum von $\inf_{(\mathbf{u}(\tau|k))} J(\mathbf{x}(k), (\mathbf{u}(\tau|k)))$ nicht eindeutig ist.

Bemerkung 13. Für den Fall, dass (3.7) ein PWA-System ist, wird in [LHWB06] für die Gewichtungsmatrizen einer Zielfunktion der Form (3.6) und für Endzustandsbeschränkungen eine Berechnungsvorschrift angegeben, unter der die Voraussetzungen des Satzes 2 erfüllt sind.

Falls der explizit, multiparametrisch berechnete Regler eines PWA- bzw. MLD-Systems ein stückweise affines Zustandsrückführungsgesetz (vgl. dazu Satz 1) über einer polyedrischen Partition der zulässigen Zustände ist, kann mit Hilfe von Analysewerkzeugen die Stabilität und Zulässigkeit für das geregelte System auch a posteriori verifiziert werden.

[MB06, Chr06] weist darauf hin, dass Stabilität und Zulässigkeit im Falle der klassischen MPC mit Echtzeitoptimierung schwer vorherzusagen und zu analysieren sind. Deshalb muss häufig die Stabilität des Reglers durch Hinzufügen von Beschränkungen an eine Endzustandsmenge im Vorschauhorizont gesichert werden (vgl. Satz 2). Dies führt unter Umständen zu einer sehr konservativen Reglereinstellung und reduziert signifikant die Menge zulässiger Zustände für die MPC. Mit dem expliziten Reglerdesign können solche Beschränkungen weggelassen werden und die Einhaltung von Beschränkungen des geregelten Systems kann auf Basis der multiparametrischen Berechnung direkt verifiziert werden. Damit kann die Konservativität und die Kompliziertheit der Reglerauslegung deutlich reduziert werden.

²Definition siehe Anhang S. 108

[MB06] argumentiert weiter, dass bereits für beschränkte lineare Systeme die Stabilität über die Nullstellen der charakteristischen Gleichung nicht mehr global gezeigt werden können. Allgemein ist dieses Problem entweder \mathcal{NP} -vollständig oder nicht entscheidbar. Weiterhin sei es meist aussichtslos, die (In-)Stabilität eines PWA-Systems aus der (In-)Stabilität seiner affinen Teilsysteme abzuleiten.

Ein weites Feld an Methoden mit variierendem Grad an Konservativität wurde zur Stabilitätsanalyse auf Basis der Lyapunov-Theorie entwickelt. Da es aber keine Standardmethode zur Konstruktion von Lyapunov-Funktionen gibt, wurden Algorithmen auf breiten Klassen von Basisfunktionen entwickelt, zum Beispiel die Berechnung einfacher quadratischer, stückweise affiner, stückweise quadratischer und stückweise polynomialer Lyapunov-Funktionen. Diese können unter anderem mit Hilfe der linearen Optimierung bestimmt werden. Die Berechnungskomplexität steigt dabei rapide mit der Ordnung des PWA-Systems und der Zahl an Dynamiken in starker Abhängigkeit von der Zahl möglicher Dynamikwechsel. Ein detaillierter Vergleich der oben genannten Techniken zur Stabilitätsanalyse wird in [BGLM05] gegeben.

Diskussion: Wie bereits in Abschnitt 3.2.3 festgestellt, birgt die Modellierung auf festem Zeitgitter die größte Fehlerquelle der MPC in sich. Dazu stellt [SE02] einen MILP-basierten MPC-Ansatz vor, der es erlaubt, die Zeitschritte – also mögliche Schaltzeitpunkte – auf Basis einer endlichen Auswahlmenge zu bestimmen. Ferner wird dort vorgeschlagen, die Linearisierung des Modells um den gerade aktuellen Systemzustand $\mathbf{x}(0|k)$ in jedem Schritt der Echtzeitoptimierung vorzunehmen. Sofern das Modell nur moderate Nichtlinearitäten besitzt, zeigt diese Methode eine Verbesserung gegenüber den bisher vorgestellten Ansätzen. Eine explizite multiparametrische Reglerberechnung ist damit nicht mehr möglich.

In der Literatur werden weitere Ansätze diskutiert. So stellt [LJ06] einen Ansatz mit zentral gerechneten suboptimalen Steuerungen vor, die lokal auf kurzem Zeithorizont nochmals optimiert werden, wodurch eine Konvergenz hin zur optimalen Steuerung erreicht wird. Auch [DS09a, DS09b] stellt ein verteiltes kommunikationsbasiertes MPC-Schema im Multiagenten-Kontext vor und [BF06] präsentiert eine hierarchisch zerlegte, heuristisch unterstützte MPC-Regelung zur Koordination von Mehrfahrzeugen in Echtzeit.

3.4 Weiterführende Konzepte

3.4.1 Iterative Verbesserung der Optimallösung

Numerische Methoden der nichtlinearen diskret-kontinuierlichen Optimalsteuerung – konkret der Ansatz mit B&B und direkter Kollokation – und die approximative Optimierung auf gemischt-ganzzahligen linearen Modellen weisen spezifische Vorzüge auf, die sich in einem kombinierten Konzept synergetisch zusammenführen lassen (vgl. Abb. 3.8). Dabei werden die eigenständigen Konzepte um eine automatisierte, zentrale Steuereinheit ergänzt, welche auf Basis heuristischer Regeln eine Bewertung der Lösung vornimmt und die daraus abgeleiteten Anweisungen zur Verbesserung der Effizienz oder Approximationsgenauigkeit an die gewählte Komponente übergibt. Dadurch kann einerseits eine Effizienzsteigerung der nichtlinearen Methode erreicht werden und andererseits die Approximationsgenauigkeit eines MILP-Modells adaptiv verbessert werden.

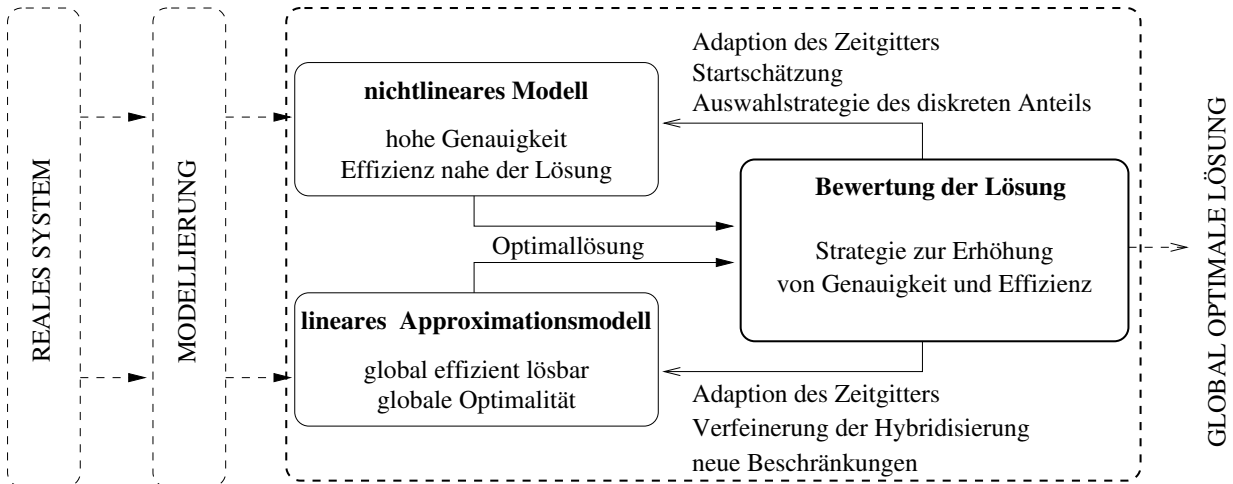


Abbildung 3.8: Iterative Verbesserung von Modell und Optimallösung.

Effizienzsteigerung nichtlinearer Methoden: Durch Vorgabe einer Auswahlstrategie für den diskreten Problemanteil kann die Effizienz der gemischt-ganzzahligen Optimierung – insbesondere im nichtlinearen Fall – gesteigert werden. Eine solche Auswahlstrategie lässt sich einerseits aus dem hierarchischen Systemmodell direkt ableiten oder kann mit Hilfe approximativer Lösungen gewonnen werden. Liegt ein B&B-Baum aus der Berechnung mit Approximationsmodell – vollständig oder teilweise – vor, so kann dieser mit all seinen Schranken und Gewichten in die Baum-Suchstrategie der diskret-kontinuierlichen direkten Kollokation einfließen. Auf Basis einer approximativen Lösung lässt sich ferner eine für die Effizienz des B&B-Verfahrens wichtige untere Schranke für das Gesamtproblem abschätzen.

Effizient berechenbare Startschätzungen auf Basis eines Approximationsmodells unterstützen die Lösung des gemischt-ganzzahligen NLP, da damit die lokale quadratische Konvergenz eines SQP-Verfahrens zum Tragen kommt.

Iterative Verbesserung der Approximationsgenauigkeit Um die in Abschnitt 3.2.3 genannten Approximationsfehler zu verringern, empfiehlt sich weiterhin eine sukzessive Modellverbesserung. Dies kann durch gezieltes Einführen neuer Zeitpunkte geschehen oder durch eine verbesserte Approximation der Dynamik und der aktiven Nebenbedingung in der Umgebung einer gefundenen Optimallösung.

Für eine erneute Berechnung mit erhöhter Approximationsgenauigkeit können auf Basis einer Optimallösung der groben Approximation untere und obere Schranken des Optimums mit verbessertem Modell geschätzt werden.

3.4.2 Anwendung gemischt-ganzzahliger MPC in kooperativen Fahrzeugsystemen

Gegenüber dem Standard-MPC-Konzept werden spezielle Anpassungen zum Einsatz in kooperativen Fahrzeugsystemen vorgestellt. Dabei werden insbesondere die dezentrale Systemstruktur mit möglicherweise instabiler Kommunikation und die Frage nach einem hinreichend langen Zeithorizont betrachtet.

MPC auf nicht-äquidistantem Zeitgitter: Um das Zuweisungsproblem kooperativer Aufgaben mit einem MPC-Ansatz zu lösen, bedarf es eines ausreichend langen Zeithorizontes $[t_{k|0}, t_{k|N}]$, so dass die zuzuweisenden Aufgaben auf Basis des Horizonts hinreichend gut zur Bewertung in der MPC-Zielfunktion berücksichtigt werden können.

Andererseits steigt mit der Zahl an Zeitdiskretisierungspunkten N die Berechnungskomplexität zur Optimierung im Allgemeinen exponentiell an. Ferner steigt mit wachsender Zeit auch die Unsicherheit in der Prädiktion, was insbesondere für Fahrzeuge in dynamischen Umgebungen (z. B. im Roboterfußball) gilt.

Als passende Konzeption wird deshalb die Verwendung eines festen, jedoch nicht äquidistanten Zeitgitters vorgeschlagen. Verlängert man die Schrittweite $\Delta_{t,\tau}$ mit wachsendem τ , trägt man der gewünschten Genauigkeit und Optimalität der aktuellen Reglerentscheidung und der Modellunsicherheit mit wachsendem τ Rechnung.

Einbindung in eine dezentrale Entscheidungsarchitektur: Die bisher vorgestellten Ansätze haben das Gesamtproblem mit vollständig zur Verfügung stehenden Informationen über alle Komponenten betrachtet. Um die vorgestellten MPC-Methoden in eine dezentrale Entscheidungsarchitektur einzubinden, bedarf es einer genauen Analyse der jeweils in einer Einheit zur Verfügung stehenden Informationen. Grundsätzlich kann jede lokale Entscheidung nur so gut sein, wie es die Informationslage und der Umfang eines internen MPC-Modells erlaubt.

Um auf Fahrzeugen mit beschränkten Rechenkapazitäten in Echtzeit eine MPC-Optimierung mit ausreichendem Zeithorizont N durchführen zu können, schlagen wir eine Modifikation des Gesamtsystem-Modells vor, das die Fahrdynamik des zu steuernden Einzelfahrzeugs in vollem Umfang verwendet, sich für alle anderen mobilen Objekte aus dessen Umwelt jedoch nur auf möglichst grobe Approximationen stützt. Damit können sowohl die vorhandenen Unbestimmtheiten und die nötige Effizienz im Einsatz auf realen Systemen beachtet werden.

Da MILP-Löser oftmals die Eigenschaft zeigen, in relativ wenigen Iterationsschritten der Optimallösung bereits sehr nahe zu kommen, lassen sich Echtzeitforderungen gegebenenfalls auch über Zeitbeschränkungen erfüllen. Der MPC-Regler verwendet dann die beste, sub-optimale Steuerung, die innerhalb eines festgelegten Zeitintervalls gefunden werden konnte.

Besteht im Mehrfahrzeugsystem die Möglichkeit zur internen Kommunikation, so bieten sich darüber hinaus verteilt arbeitende Ansätze an, in denen jede Entscheidungseinheit Teillösungen für ein gemeinsames Gesamtproblem einbringt (siehe dazu z.B. [BCM09]).

Zusammenfassung des Kapitels

Aufbauend auf hybride dynamische Systemmodelle (Kapitel 2) wurde die numerische Behandlung von Optimalsteuerungsproblemen diskutiert. Insbesondere wurde für die in Abschnitt 2.2.2 vorgeschlagene hierarchische Erweiterung hybrider Zustandsautomaten die Transformation in eine gemischt-ganzzahlige Formulierung durchgeführt, so dass sowohl diskrete als auch kontinuierliche Zustands- und Steuervariablen simultan durch Optimierung bestimmt werden können.

Es wurden Methoden zur Behandlung allgemeiner, nichtlinearer, diskret-kontinuierlicher Probleme beleuchtet und die methodische Brücke über eigene Approximationskonzepte

hin zu linearen Optimierungsansätzen geschlagen. Die Anwendung MILP-basierter MPC-Regler wurde diskutiert und für eine neuartige Anwendung in der Zuweisung kooperativer, mobiler Aufgaben (siehe Abschnitt 5.3) eine Grundlage geschaffen.

Mit weiterführenden Konzepten der MPC-Anwendung auf nicht äquidistantem Gitter, der Dezentralisierung und der iterativen Verbesserung der Approximationsgenauigkeit wurden die allgemeinen Betrachtungen zur diskret-kontinuierlichen Optimalsteuerung abgeschlossen.

4 Modellierung kooperativer Mehrfahrzeugsysteme

Kapitelinhalt:

Ein eigenes, breit anwendbares Modellierungskonzept wird vorgestellt, das Modelle hierarchischer hybrider Automaten mit abgebildeter Fahrdynamik und Beschränkungen – auf Basis der bisher beleuchteten Grundlagen – der Optimierung zugänglich macht. Dabei werden unter anderem Kommunikationsbedingungen, unterlagerte veränderliche Graphenstrukturen und das Einsatzgebiet beschrieben. Für verschiedene Teilaspekte werden neuartige MILP-Formulierungen vorgeschlagen. Insbesondere wird auf die Vorzüge einer neuartigen Erweiterung der skalierbaren Beschreibung zur Hindernis- und Kollisionsvermeidung sowie auf die Behandlung von Konnektivitätsbedingungen in Drahtlossensornetzwerken hingewiesen.

Sofern nicht anders angegeben, wird folgende Notation im Welt-Koordinatensystem verwendet:

$(x^i(k), y^i(k))$ bezeichnet die Position in kartesischen Koordinaten (in \mathbb{R}^2) von Fahrzeug i zum Zeitpunkt t_k . $(v_x^i(k), v_y^i(k))$ steht für den zugehörigen Geschwindigkeitsvektor in x - und y -Richtung.

4.1 Allgemeines Modellierungskonzept

Auf Basis der Vorarbeiten in den Kapiteln 2 und 3 wird ein eigenes, allgemeines Modellierungskonzept vorgestellt, mit Hilfe dessen sich ein umfassendes, approximatives, gemischt-ganzzahliges Modell für ein allgemeines kooperatives Mehrfahrzeugsystem erstellen lässt.

Ausgangspunkt der Modellierung (vgl. Abb. 4.1) stellt die (hierarchische) diskrete Grundstruktur des Systems dar, die als Grundstruktur eines hierarchischen hybriden Automaten (vgl. 2.2.2) beschrieben wird.

Der anschließende Schritt beschreibt für alle Fahrzeuge die Dynamik und deren Beschränkungen (Abschnitt 4.2). Entscheidet man sich hier für eine approximative Modellierung (vgl. 3.2.1), so führt eine Hybridisierung über eine PWA-Beschreibung gegebenenfalls zu einer erweiterten diskreten Problemstruktur und zu Anpassungen des Automatengerüsts.

Nachdem das dynamische Grundverhalten vollständig abgebildet ist, werden Beschränkungen der Koordination – beispielsweise zur Kollisionsvermeidung (Abschnitt 4.3.1) – und eine Zielfunktion (Abschnitt 4.2.3) zur späteren Minimierung in das Problem aufgenommen. Auch hier kann eine Approximation zu einer Ausweitung der diskreten Struktur führen. Sowohl Zielfunktion, als auch Beschränkungen werden weiterhin dazu verwendet, die gestellten kooperativen Aufgaben zu beschreiben.

Ein für die Effizienz der anschließenden Optimierung wichtiger Schritt besteht in der strukturellen Modellverbesserung. Dies kann durch Anwendung zusätzlichen Wissens über eine Optimallösung oder systematisch durch Erreichbarkeitsanalysen erfolgen, zum Beispiel automatisiert mit Methoden der Logikprogrammierung CLP. Lässt sich ermitteln, dass bestimmte diskrete Zustände zu bestimmten Zeitpunkten nicht erreichbar sind, so erweist es sich aus Sicht der gemischt-ganzzahligen Optimierung als effizienzverbessernd, dazu einfache Nebenbedingungen der Art „ $b^i(k) = 0$ “ aufzunehmen. Weitere allgemeine Empfehlungen der empfindlichen Formulierung gemischt-ganzzahliger Optimierungsprobleme werden in [Kal02, Kap. 5 „Die Kunst guter Modellierung“] vorgestellt.

Der Modellierungsprozess spielt zur Anwendung von Verfahren der gemischt-ganzzahligen Optimierung eine zentrale, sensible Rolle. Bereits kleine Modifikationen in der Beschreibung des Steuer-Zustands-Raumes können dazu führen, dass sich die Dauer zur Bestimmung eines Optimums um Größenordnungen ändert [Kal02].

Im speziellen Anwendungsfall können einzelne Teilschritte des Konzeptes übergangen werden. Beim Aufbau eines approximativen Ersatzmodells wird die Approximation der Ausdrücke in jedem Schritt unmittelbar mit durchgeführt.

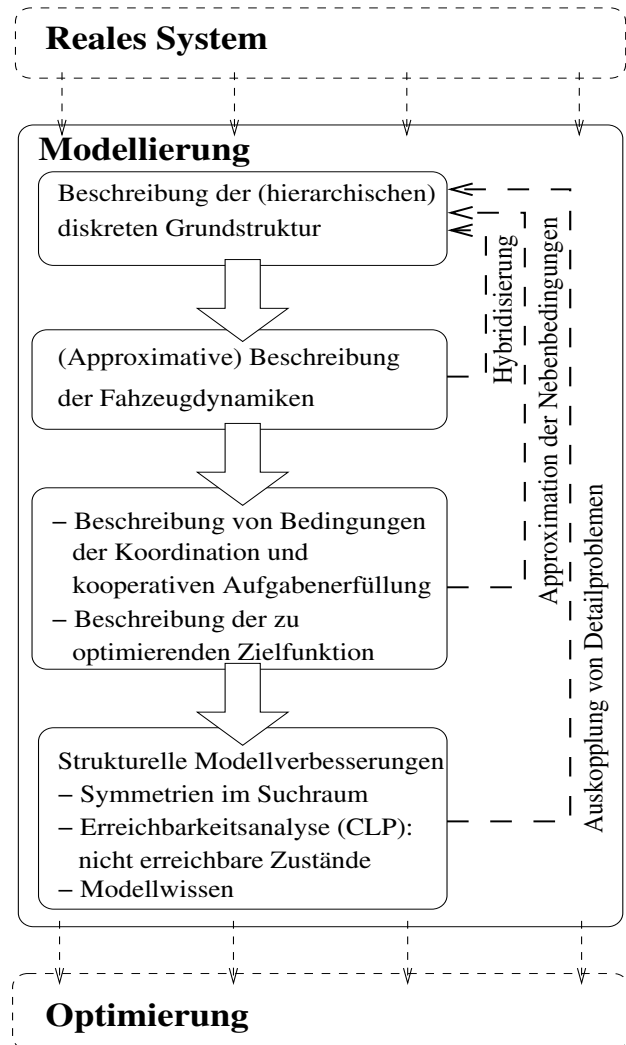


Abbildung 4.1: Modellierungskonzept: ein System wird Optimierungsmethoden zugänglich gemacht.

4.2 Modellierung und Bewertung kooperativer Mobilität

4.2.1 Fahrzeugdynamik

Es wird eine Gruppe von n_v kooperierenden Fahrzeugen betrachtet, welche in einem Gebiet $G \subset \mathbb{R}^2$ agiert. Die entwickelten Prinzipien lassen sich problemlos auf Bewegungen im Raum ($G \subset \mathbb{R}^3$) übertragen.

Ziel ist es, die Fahrzeugdynamik eines kooperativen Mehrfahrzeugsystems so zu modellieren, dass einerseits die physikalische Bewegung hinreichend genau abgebildet wird und andererseits das resultierende Modell der numerischen Optimalsteuerung zugänglich ist.

Gesucht sind Fahrzeugsteuerungen, so dass – unter Einhaltung der Bewegungsdifferentialgleichungen und physikalischen Beschränkungen – ein gewünschtes Systemverhalten (zum Beispiel kooperative Aufgabenerfüllung, Formationswechsel, taktisches Mannschaftsspiel) optimal erreicht wird.

Bewegungsdynamik eines Einzelfahrzeugs: In der allgemeinsten hier betrachteten Form lässt sich die Bewegungsdynamik eines einzelnen Fahrzeugs i durch

$$\dot{\mathbf{x}}^i(t) = \mathbf{f}_q^i(\mathbf{x}^i(t), \mathbf{u}^i(t)) \quad (4.1)$$

beschreiben, wobei $\mathbf{x}^i(t) \in \mathbb{R}^{n_{x,i}}$ den kontinuierlichen Systemzustand (z.B. Position, Geschwindigkeit, Orientierung, ...), $\mathbf{u}^i(t) \in \mathbb{R}^{n_{u,i}}$ die kontinuierlichen Steuervariablen (z.B. Beschleunigung, Lenkwinkel, ...) und q einen bestimmten Bewegungsmodus (z.B. Fahren mit / ohne Last) repräsentieren. Abhängig vom Detaillierungsgrad des physikalischen Modells reichen diese Bewegungsdifferentialgleichungen allgemein von einfach kinematischen bis hin zu komplexen nichtlinearen Fahrzeugdynamikmodellen. Zur Untersuchung der Kernproblematik kooperativer Mobilität beschränken wir uns auf einfache Modelle aus der Literatur, wie sie in Tabelle 4.1 aufgelistet sind.

Modell	Dynamik	Erläuterung
Punktmasse (1. Ordnung)	$\dot{x} = u_x$ $\dot{y} = u_y$	Fahrzeug besitzt keine Orientierung; Steuerung: Geschwindigkeit in x -, y -Richtung
Punktmasse (2. Ordnung)	$\dot{x} = v_x$ $\dot{y} = v_y$ $\dot{v}_x = u_x$ $\dot{v}_y = u_y$	Fahrzeug besitzt keine Orientierung; Steuerung: Beschleunigung in x -, y -Richtung (vgl. Abb. 4.2b)
Differentialantrieb	$\dot{x} = \frac{r}{2}(u_l + u_r) \sin(\theta)$ $\dot{y} = \frac{r}{2}(u_l + u_r) \cos(\theta)$ $\dot{\theta} = \frac{r}{L}(u_r - u_l)$	Steuerung: Umdrehungsgeschwindigkeiten der Räder (vgl. Abb. 4.2a)
Einrad-Modell	$\dot{x} = u_v \sin(\theta)$ $\dot{y} = u_v \cos(\theta)$ $\dot{\theta} = u_\theta$	Steuerung: Geschwindigkeit u_v und Winkelgeschwindigkeit u_θ Abgeleitet aus dem Modell eines Differentialantriebs (vgl. Abb. 4.2a)
Einrad-Modell (Dubin-car)	$\dot{x} = v \sin(\theta)$ $\dot{y} = v \cos(\theta)$ $\dot{v} = u_v$ $\dot{\theta} = u_\theta$	Steuerung: Beschleunigung u_v und Winkelbeschleunigung u_θ

Tabelle 4.1: Einfache Bewegungsdynamiken zur Untersuchung kooperativer Mobilität

Über Transformationen und Approximationen lassen sich damit viele rad- und kettengetriebene Fahrzeuge mit den genannten Modellen identifizieren, so dass eine hinreichend genaue Basis zur Abschätzung und Planung kooperativer Mobilität vorliegt.

Der Anfangszustand eines Fahrzeugs $\mathbf{x}^i(0) = \mathbf{x}_0^i$ zur Startzeit $t_0 = 0$ kann im Allgemeinen genauso wie die Endzeit t_f und die Endzustände $\mathbf{x}^i(t_f)$ variabel oder fest ($\mathbf{x}^i(0) = \mathbf{x}_0^i$, $\mathbf{x}^i(t_f) = \mathbf{x}_f^i$) gewählt sein.

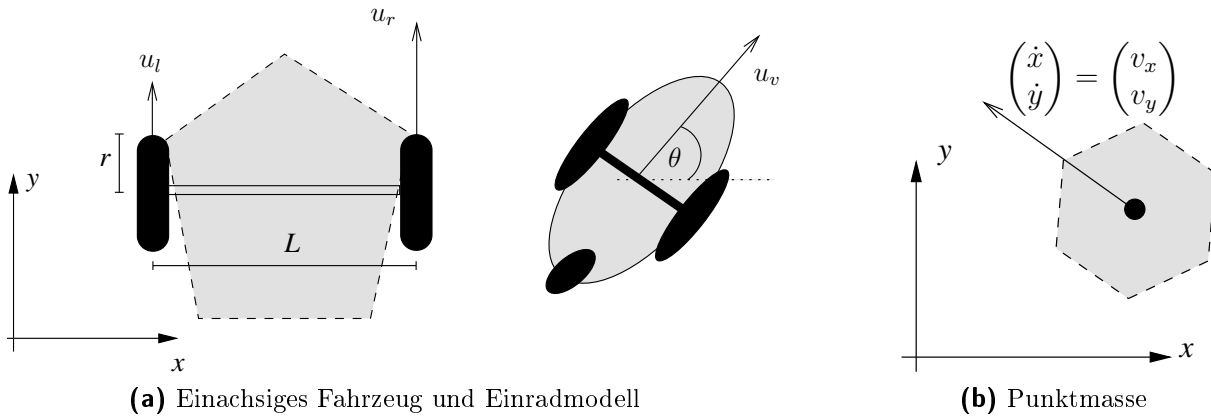


Abbildung 4.2: Schematische Darstellung der Fahrzeugmodelle.

Zustände und Steuervariablen einzelner Fahrzeuge unterliegen Beschränkungen, die sich aus der Physik des Fahr- oder Flugzeugs (z.B. maximale Bewegungsradien, Höchstgeschwindigkeiten) und dessen Umgebung ergeben; in allgemeiner Form treten diese als

$$\mathbf{g}_q^i(\mathbf{x}^i(t), \mathbf{u}^i(t)) \leq 0 \quad (4.2)$$

auf. Für das wichtige nichtlineare Detail der Modellierung minimaler Kurvenradien wird im Folgenden eine lineare zeitdiskrete Formulierung entwickelt, die unabhängig von der Fahrzeugdynamik angewendet werden kann.

Lineare, zeitdiskrete Modellierung beschränkter Kurvenradien: Der Blick in die Literatur (z.B. [RH02]) zeigt, dass dieser Aspekt in MILP-Modellen oft umgangen wird, da es sich um eine nichtlineare Beschränkung handelt, die mit gängigen linearisierten Formulierungen zu einer unverhältnismäßig großen diskreten Struktur im Modell führt.

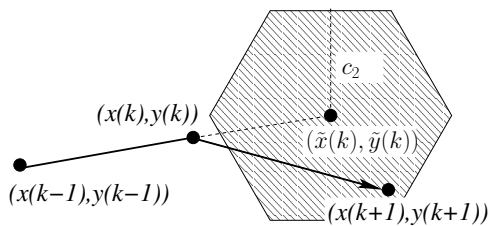


Abbildung 4.3: Veranschaulichung zur Beschränkung von Kurvenradien.

Es wird deshalb ein Ansatz vorgeschlagen, der ohne weitere binäre Variablen auskommt und für viele Fahrdynamiken eine deutlich verbesserte Modellgenauigkeit mit sich bringt (siehe dazu auch die Anwendung in Abschnitt 5.1.2). Die Idee dabei (vgl. Abb. 4.3) besteht darin, das Modell um Hilfsvariablen $(\tilde{x}^i(k), \tilde{y}^i(k))$ zu erweitern, in deren Nähe der nächste Wegpunkt liegen muss. Dabei liegt $(\tilde{x}^i(k), \tilde{y}^i(k))$ in der Verlängerung des Weges von $(x^i(k-1), y^i(k-1))$ nach $(x^i(k), y^i(k))$. Formal lässt sich schreiben

$$\tilde{x}^i(k) = x^i(k) + c_1 \Delta_t (x^i(k) - x^i(k-1)), \quad (4.3a)$$

$$\tilde{y}^i(k) = y^i(k) + c_1 \Delta_t (y^i(k) - y^i(k-1)), \quad (4.3b)$$

$$\|(\tilde{x}^i(k+1), \tilde{y}^i(k+1)) - (x^i(k+1), y^i(k+1))\| \leq c_2, \quad (4.3c)$$

mit Konstanten c_1 und c_2 . Diese sind so zu wählen, dass die Beschleunigung und Verzögerung in Geradeaus-Richtung nicht über bereits vorliegende Beschränkungen hinaus begrenzt wird.

Implizit berücksichtigt diese Formulierung, dass der minimale Kurvenradius mit wachsender Momentangeschwindigkeit steigt. Dies korreliert mit der Notwendigkeit, die Haftreibung in der Kurvenfahrt nicht zu verlieren. Die lineare Approximation von Normen in (4.3c) wird im Abschnitt 4.3 genauer beleuchtet werden.

Modell des Gesamtsystems: Die Berechnung optimaler Steuerungen und Trajektorien wird wesentlich komplexer beim Übergang von einem Einzelfahrzeug- zu einem Mehrfahrzeugsystem mit kooperativen Aufgaben.

Die Bewegungsdynamik des Gesamtsystems ergibt sich zunächst aus den Dynamiken der einzelnen Fahrzeuge

$$\dot{\mathbf{x}} := \begin{pmatrix} \dot{\mathbf{x}}^1 \\ \vdots \\ \dot{\mathbf{x}}^{n_v} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_q^1(\mathbf{x}^1(t), \mathbf{u}^1(t)) \\ \vdots \\ \mathbf{f}_q^{n_v}(\mathbf{x}^{n_v}(t), \mathbf{u}^{n_v}(t)) \end{pmatrix} = \mathbf{f}_q(\mathbf{x}, \mathbf{u}), \quad (4.4)$$

mit $q \in \mathbb{Q}$ und $(\mathbf{u}^1, \dots, \mathbf{u}^{n_v})^T =: \mathbf{u}$ Die Komponenten des Systems sind durch algebraische Beschränkungen eng aneinander gekoppelt; entweder durch physikalische Notwendigkeiten, wie die Kollisionsvermeidung oder durch eine gemeinsame Aufgabe.

4.2.2 Zeitdiskrete, lineare Formulierung einfacher Fahrdynamiken

Auf die vorgestellten einfachen Fahrzeugdynamiken (S. 53), werden die in den Abschnitten 2.3 und 3.2 entwickelten Methoden angewendet und resultierende zeitdiskrete, lineare Beschreibungen der Form

$$\mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) + \mathbf{H}\mathbf{u}(k) + \mathbf{\phi} \quad (4.5)$$

angeben.

Punktmassenmodell 1. Ordnung: Die einfachste betrachtete Beschreibung einer Bewegung in der Ebene ist durch das Dynamikmodell

$$\begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \end{pmatrix} = \begin{pmatrix} u_x(t) \\ u_y(t) \end{pmatrix} \quad (4.6)$$

gegeben. Das zugehörige zeitdiskrete Modell (4.5) ergibt sich bei Linearisierung um $(x, y) = (x_0, y_0)$ und $(u_x, u_y) = (0, 0)$ sowohl mit Euler-Diskretisierung, als auch mit exakter Diskretisierung, zu

$$\mathbf{\Phi} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} \Delta_t \\ \Delta_t \end{pmatrix}, \quad \mathbf{\phi} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \quad (4.7)$$

Die Linearisierungen sind nicht durch eine Zustandsgröße parametrisiert und besitzen somit globale Gültigkeit im gesamten Zustandsraum. Eine weitere Hybridisierung nach Abschnitt 3.2.1 ist somit nicht nötig. Der Approximationsfehler ergibt sich allein durch die Annahme, dass die Steuerung \mathbf{u} auf einem festen Zeitgitter stückweise konstant sei.

Punktmassenmodell 2. Ordnung: Eine detailliertere Beschreibung einer Fahrdynamik erlaubt das Doppelintegrator-Modell

$$\begin{pmatrix} \ddot{x}(t) \\ \ddot{y}(t) \end{pmatrix} = \begin{pmatrix} \dot{v}_x(t) \\ \dot{v}_y(t) \end{pmatrix} = \begin{pmatrix} u_x(t) \\ u_y(t) \end{pmatrix}. \quad (4.8)$$

Die zugehörigen zeitdiskreten Beschreibungen (4.5) ergeben sich wie folgt:

- Linearisierung um $(x, v_x, y, v_y) = (0, 0, 0, 0)$ und $(u_x, u_y) = (0, 0)$ mit Euler-Diskretisierung:

$$\mathbf{\Phi} = \begin{pmatrix} 1 & \Delta_t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta_t \\ 0 & 0 & 0 & 1 \end{pmatrix}, \mathbf{H} = \begin{pmatrix} 0 & 0 \\ \Delta_t & 0 \\ 0 & 0 \\ 0 & \Delta_t \end{pmatrix}, \phi = \mathbf{0} \quad (4.9)$$

- Linearisierung um $(x, v_x, y, v_y) = (0, 0, 0, 0)$ und $(u_x, u_y) = (0, 0)$ mit exakter Diskretisierung:

$$\mathbf{\Phi} = \begin{pmatrix} 1 & \Delta_t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta_t \\ 0 & 0 & 0 & 1 \end{pmatrix}, \mathbf{H} = \begin{pmatrix} \frac{1}{2}\Delta_t^2 & 0 \\ \Delta_t & 0 \\ 0 & \frac{1}{2}\Delta_t^2 \\ 0 & \Delta_t \end{pmatrix}, \phi = \mathbf{0} \quad (4.10)$$

Auch hier gilt: Die Linearisierungen sind nicht durch eine Zustandsgröße parametrisiert und besitzen somit globale Gültigkeit im gesamten Zustandsraum. Eine weitere Hybridisierung (siehe Abschnitt 3.2.1) ist somit nicht nötig. Der Approximationsfehler ergibt sich allein durch die Einschränkung, dass die Steuerung \mathbf{u} auf einem festen Zeitgitter als stückweise konstant angenommen wird.

Fahrzeug mit Differentialantrieb: Zur Dynamik

$$\begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{pmatrix} = \begin{pmatrix} \frac{r}{2}(u_l(t) + u_r(t)) \sin(\theta(t)) \\ \frac{r}{2}(u_l(t) + u_r(t)) \cos(\theta(t)) \\ \frac{r}{L}(u_r(t) - u_l(t)) \end{pmatrix} \quad (4.11)$$

lauten die zugehörigen zeitdiskreten Modelle (4.5):

- Linearisierung um $(x, y, \theta) = (x_0, y_0, \theta_0)$ und $(u_l, u_r) = (0, 0)$, sowohl mit Euler-Diskretisierung als auch mit exakter Diskretisierung:

$$\mathbf{\Phi} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \mathbf{H} = \begin{pmatrix} \frac{r}{2}\Delta_t \sin(\theta_0) & \frac{r}{2}\Delta_t \sin(\theta_0) \\ \frac{r}{2}\Delta_t \cos(\theta_0) & \frac{r}{2}\Delta_t \cos(\theta_0) \\ \frac{-r\Delta_t}{L} & \frac{r\Delta_t}{L} \end{pmatrix}, \phi = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (4.12)$$

Die Dynamik (4.11) ist nichtlinear in θ , was zur Folge hat, dass eine linearisierte Lösung nur lokal in der Umgebung um θ_0 eine brauchbare Näherung der Fahrzeugdynamik darstellt.

- Mittels $u_v := \frac{r}{2}(u_l + u_r)$ und $u_\theta := \frac{r}{L}(u_r - u_l)$ lässt sich ein Fahrzeug mit Differentialantrieb auch in der numerisch günstigeren Formulierung eines Einrad-Modells beschreiben. Für die optimale Hybridisierung nach Abschnitt 3.2.1 verweisen wir deshalb auf folgende Betrachtungen.

Einrad-Modell: Zur Dynamik

$$\begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{pmatrix} = \begin{pmatrix} u_v(t) \sin(\theta(t)) \\ u_v(t) \cos(\theta(t)) \\ u_\theta(t) \end{pmatrix} \quad (4.13)$$

ergeben sich die zugehörigen zeitdiskreten Beschreibungen (nach 4.5) wie folgt:

- Linearisierung um $(x, y, \theta) = (x_0, y_0, \theta_0)$ und $(u_v, u_\theta) = (0, 0)$, sowohl mit Euler-Diskretisierung als auch mit exakter Diskretisierung:

$$\mathbf{\Phi} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \mathbf{H} = \begin{pmatrix} \Delta_t \sin(\theta_0) & 0 \\ \Delta_t \cos(\theta_0) & 0 \\ 0 & \Delta_t \end{pmatrix}, \mathbf{\phi} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (4.14)$$

Auch hier gilt: Die Dynamik (4.13) ist nichtlinear in θ . Die linearisierte Lösung stellt nur lokal in der Umgebung um θ_0 eine hinreichend genaue Näherung der Fahrzeugdynamik dar.

- Zur optimalen Hybridisierung nach Abschnitt 3.2.1 wird nur der nichtlineare Anteil betrachtet,

$$\hat{\mathbf{f}} : [0, u_{max}] \times [0, 2\pi) \rightarrow [-u_{max}, u_{max}]^2 : \hat{\mathbf{f}}(u, \theta) = \begin{pmatrix} u_v \sin(\theta) \\ u_v \cos(\theta) \end{pmatrix}.$$

Die Minimierung der Gleichung (3.4) erfolgt mit Simplizes der Menge $\mathbb{M}_q = \{M_1, M_2, \dots, M_L\}$, wie sie in Abb. 4.4 dargestellt sind. Die mit „*“ bezeichneten Koordinaten der Simplizes werden wie auch alle zugehörigen Stützpunkte $z(\theta^l, u^l)$, durch die Lösung eines nichtlinearen beschränkten Optimierungsproblems (3.4) bestimmt.

Da für die optimale PWA-Formulierung die Periodizität von $\hat{\mathbf{f}}$ erhalten bleiben soll, wird die Gleichungsbeschränkung $z(0, 0) = z(2\pi, 0)$ und $z(0, u_{max}) = z(2\pi, u_{max})$ in das Optimierungsproblem aufgenommen. Die sich für diesen Fall ergebende Lösung ist in den Graphiken 4.4 abgebildet.

Einrad-Modell (Dubin-car): Die erweiterte Einrad-Dynamik

$$\begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{v}(t) \\ \dot{\theta}(t) \end{pmatrix} = \begin{pmatrix} v(t) \sin(\theta(t)) \\ v(t) \cos(\theta(t)) \\ u_v(t) \\ u_\theta(t) \end{pmatrix} \quad (4.15)$$

besitzt folgende zeitdiskrete Approximation (4.5):

- Linearisierung um $(x, y, v, \theta) = (x_0, y_0, v_0, \theta_0)$ und $(u_v, u_\theta) = (0, 0)$ mit Euler-Diskretisierung:

$$\mathbf{\Phi} = \begin{pmatrix} 1 & 0 & \Delta_t \sin(\theta_0) & v_0 \Delta_t \cos(\theta_0) \\ 0 & 1 & \Delta_t \cos(\theta_0) & -v_0 \Delta_t \sin(\theta_0) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \mathbf{H} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \Delta_t & 0 \\ 0 & \Delta_t \end{pmatrix}, \mathbf{\phi} = \begin{pmatrix} v_0 \Delta_t \sin(\theta_0) \\ v_0 \Delta_t \cos(\theta_0) \\ 0 \\ 0 \end{pmatrix} \quad (4.16)$$

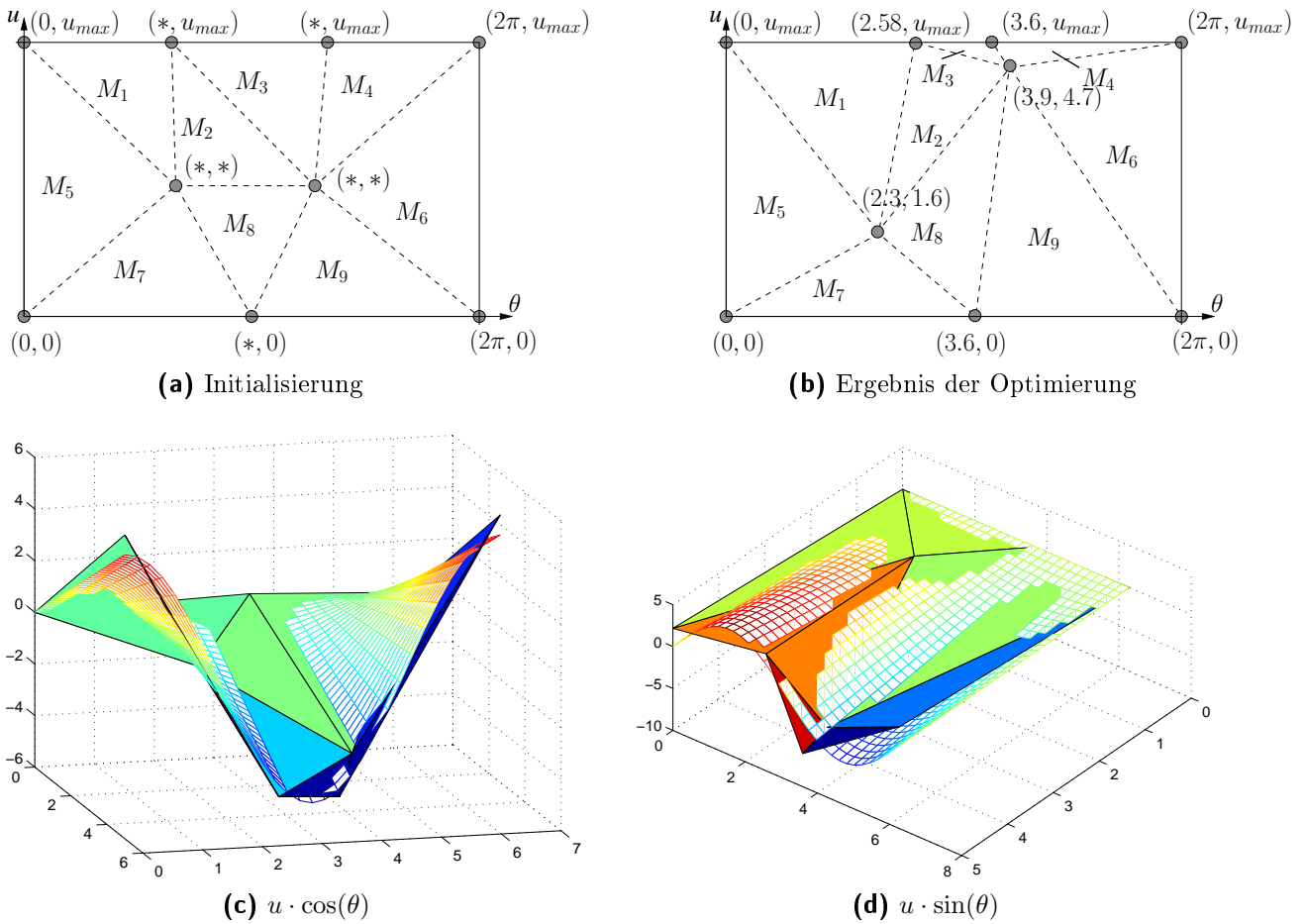


Abbildung 4.4: Einteilung des Steuer-Zustands-Raumes (oben) und berechnete optimale Hybridisierung (in Flächendarstellung) und Originalfunktion (in Gitterdarstellung).

- Linearisierung um $(x, y, \theta) = (x_0, y_0, v_0, \theta_0)$ und $(u_v, u_\theta) = (0, 0)$ mit exakter Diskretisierung: (Φ wie in (4.16))

$$H = \begin{pmatrix} \frac{1}{2} \Delta_t^2 \sin(\theta_0) & \frac{v_0}{2} \Delta_t^2 \cos(\theta_0) \\ \frac{1}{2} \Delta_t^2 \cos(\theta_0) & -\frac{v_0}{2} \Delta_t^2 \sin(\theta_0) \\ \Delta_t & 0 \\ 0 & \Delta_t \end{pmatrix}, \quad \phi = \begin{pmatrix} v_0 \Delta_t \sin(\theta_0) \\ v_0 \Delta_t \cos(\theta_0) \\ 0 \\ 0 \end{pmatrix} \quad (4.17)$$

- optimale Hybridisierung nach Abschnitt 3.2.1: Zur Hybridisierung wird nur der nicht-lineare Anteil betrachtet,

$$\hat{\mathbf{f}} : [0, v_{max}] \times [0, 2\pi) \rightarrow [-v_{max}, v_{max}]^2 : \hat{\mathbf{f}}(v, \theta) = \begin{pmatrix} v \sin(\theta) \\ v \cos(\theta) \end{pmatrix}.$$

Formulierung und Lösung des Minimierungsproblems zur Berechnung der optimalen PWA-Beschreibung deckt sich mit dem Vorgehen beim einfachen Einradmodell.

In Kapitel 5 werden die vorgestellten Dynamiken in verschiedenen Anwendungsbeispielen auf ihre Eignung zur Berücksichtigung für die gemischt-ganzzahlige Optimierung getestet.

4.2.3 Bewertung kooperativer Bewegung durch eine Zielfunktion

Allgemein wird die Formulierung einer Zielfunktion (vgl. Gleichung (3.1))

$$J[\mathbf{u}, q] := \Phi_{q_{n_c+1}}(\mathbf{x}(t_f)) + \sum_{k=0}^{n_c} \Phi_{q_k}(\mathbf{x}(t_{k+1}-0), q_{k+1}, \mathbf{x}(t_{k+1}+0)) \\ + \sum_{k=0}^{n_c} \int_{t_k}^{t_{k+1}} L_{q_k}(\mathbf{x}(t), \mathbf{u}(t), q) dt$$

diskutiert, mit der die Qualität einer kooperativen Aufgabenerfüllung gemessen wird.

Mit $\Phi_{q_k}(\mathbf{x}(t_{k+1}-0), q_{k+1}, \mathbf{x}(t_{k+1}+0))$ lassen sich dabei Schaltkosten an diskreten Zustandsübergängen bewerten. Die Funktion $L_{q_k}(\mathbf{x}(t), \mathbf{u}(t))$ misst die Laufkosten des sich kontinuierlich entwickelnden Systems in der Phase $[t_k, t_{k+1}]$.

Unterschieden werden zwei Fälle:

1. Die Aufgabenerfüllung selbst wird als Optimierungskriterium formuliert oder
2. die Minimierung von Kosten einer erzwungenen Aufgabe wird bewertet.

Im Fall 1 misst die Zielfunktion den Abstand zu einem idealen Zustand bzw. einer Zustandsmenge. Ziel der Optimalsteuerung ist es, diejenige Steuerung zu berechnen, mit der man dem gewünschten Endzustand möglichst nahe kommt. Ein Beispiel dafür ist die Berechnung einer optimalen Taktik im Roboterfußball. Der Idealzustand formuliert dabei beispielsweise eine Situation mit Ball im Tor und maximalem Abstand zum Gegenspieler. Die Formulierung der Idealzustandsmenge, und damit die Zielfunktion, ist allgemein sehr aufgaben- und systemspezifisch. Jede Teilmenge des diskret-kontinuierlichen Steuerzustands-Raumes kann dafür gewählt werden.

Im Fall 2 ist die Aufgabe als Zwangsbedingung formuliert und die Optimierung berechnet eine Steuerung, welche diese optimal erfüllt. Ein Beispiel dafür ist die kooperative Erkundung mehrerer Wegpunkte. Gesucht werden diejenigen Trajektorien, so dass alle Wegpunkte bei minimalem Energieeinsatz erkundet werden.

Folgende Spezialfälle werden häufig zur Bewertung der Optimalität verwendet.

Energieminimierung: Insbesondere bei Luftfahrzeugen ist die energieminimierende Planung von Missionen von besonderer Bedeutung, um das Gewicht von Energiespeichern reduzieren oder die Operationszeit verlängern zu können. Eine typische Formulierung ist es, hierbei mit

$$J = \sum_{i=1}^{n_v} m_i \cdot \sum_{k=1}^{n_c} \int_{t_{k-1}}^{t_k} \mathbf{u}^i(t)^T \mathbf{u}^i(t) dt \quad (4.18)$$

den nötigen Energieeintrag in das System zu minimieren. Dabei stellt m_i die jeweilige Fahrzeugmasse und \mathbf{u}^i die zugehörige Beschleunigung dar.

Minimierung der Gesamtzeit: Ist man daran interessiert, eine Mission in möglichst kurzer Zeit abzuschließen, wählt man eine Zielfunktion der Form

$$J = t_f + \epsilon \sum_{i=1}^{n_v} m_i \cdot \sum_{k=1}^{n_c} \int_{t_{k-1}}^{t_k} \mathbf{u}^i(t)^T \mathbf{u}^i(t) dt \quad (4.19)$$

mit $\epsilon = 0$. Bei einem klein gewählten, positiven $\epsilon > 0$ garantiert der zweite Summand eine Eindeutigkeit und „Glättung“ der Lösung. Für $\epsilon = 0$ schöpft im Allgemeinen eines der Fahrzeuge die minimale Zeit zur Aufgabendurchführung genau aus, während die Trajektorien der andern nicht eindeutig bestimmt sind. Mit einem kleinen $\epsilon > 0$ kann dem Rechnung getragen werden, ohne sukzessive ein weiteres Optimalsteuerungsproblem zu lösen. Zeit- und Energieminimierung sind in der Regel gegenläufige Ziele.

4.3 Paarweise Beziehungen und Formationsstrukturen

Dieser Abschnitt stellt die Modellierung geometrischer Strukturen vor, die als Kombination paarweiser Beziehungen und diskreter Graphenstrukturen verstanden wird. Im Kontext kooperierender Mehrfahrzeugsysteme werden damit Bedingungen der Kollisionsvermeidung, Formationsforderungen oder Voraussetzungen für eine stabile Funkkommunikation der Fahrzeuge beschreiben. Diese Aspekte sind bedeutsam bei der flächendeckenden Erkundung oder der mobilitätsgestützten Datenübertragung mit Hilfe kooperierender Fahrzeuge.

4.3.1 Paarweise Beziehungen

Paarweise Beziehungen werden betrachtet, die für ein koordiniertes und kooperatives Verhalten des Mehrfahrzeugsystems nötig sind. Wir beschränken uns darauf, die Positionen (x^{i_1}, y^{i_1}) und (x^{i_2}, y^{i_2}) zweier Fahrzeuge i_1 und i_2 zueinander in Bezug zu setzen. Das Vorgehen lässt sich aber auf allgemeine Zustands- und Steuergrößen übertragen.

Konkret werden Normen des Vektors $\mathbf{d}_{i_1, i_2} := ((x^{i_1}, y^{i_1}) - (x^{i_2}, y^{i_2}))^T$ betrachtet. Dabei spielen insbesondere

- die 1-Norm: $\|\mathbf{d}_{i_1, i_2}\|_1 = |x^{i_1} - x^{i_2}| + |y^{i_1} - y^{i_2}|$,
- die 2-Norm: $\|\mathbf{d}_{i_1, i_2}\|_2 = \sqrt{(x^{i_1} - x^{i_2})^2 + (y^{i_1} - y^{i_2})^2}$ und
- die ∞ -Norm: $\|\mathbf{d}_{i_1, i_2}\|_\infty = \max\{|x^{i_1} - x^{i_2}|, |y^{i_1} - y^{i_2}|\}$

eine wichtige Rolle. Es lassen sich damit Hindernis- und Kollisionsvermeidungsbedingungen der Form $\|\mathbf{d}_{i_1, i_2}\| \geq D_2$ formulieren oder Beziehungen innerhalb einer Formation beschreiben, $\|\mathbf{d}_{i_1, i_2}\| < D_1$.

Als lineare Näherung von $\|\mathbf{d}_{i_1, i_2}\| < D_1$ dient folgende Formulierung: ($l = 1, \dots, n_d$)

$$(x^{i_1} - x^{i_2}) \sin\left(\frac{l 2\pi}{n_d} + \varphi_0\right) + (y^{i_1} - y^{i_2}) \cos\left(\frac{l 2\pi}{n_d} + \varphi_0\right) < D_1. \quad (4.20)$$

Bemerkung 14. Mit $n_d = 4$, $\varphi_0 = 0$ wird exakt die 1-Norm beschreiben; mit $n_d = 4$ und $\varphi_0 = \frac{\pi}{4}$ die ∞ -Norm.

Durch das Einbringen von Formulierungen der Form $\|d_{i_1, i_2}\| \geq D_2$ in das Modell und somit in die Nebenbedingungen eines Optimierungsproblems ergibt sich für die Steuerungs-Zustands-Menge $\mathbb{U} \times \mathbb{X}$ eine Nichtkonvexität. In der Optimierung führen Nichtkonvexitäten im Suchraum zu Schwierigkeiten bei der Suche globaler Optima. Klassische gradientenbasierte Verfahren können trotz einer gegebenenfalls konvexen Zielfunktion keine Globalität des besten erreichten Wertes mehr garantieren.

Es lassen sich Bedingungen der Form $\|d_{i_1, i_2}\| \geq D_2$ jedoch approximativ durch eine gemischt-ganzzahlige Formulierung mit $n_d \geq 4$ festen linearen Beschränkungen ($l = 1, \dots, n_d, b_l \in \{0, 1\}$) schreiben:

$$\left[b_l = 1 \Rightarrow (x^{i_1} - x^{i_2}) \sin\left(\frac{l2\pi}{n_d} + \varphi_0\right) + (y^{i_1} - y^{i_2}) \cos\left(\frac{l2\pi}{n_d} + \varphi_0\right) \geq D_2 \right] \quad (4.21a)$$

$$\sum_{l=1}^{n_d} b_l \geq 1 \quad (4.21b)$$

Der Ausdruck (4.21a) wird schließlich mit den Methoden aus Abschnitt 2.1.2 in einen algebraischen Ausdruck übersetzt. Damit die gewünschte Eigenschaft hergestellt, im relaxierten Fall ($b_l \in [0, 1]$) ein konvexes Gebiet zu beschreiben.

In vielen kooperativen Szenarien sind die bisher beschriebenen Nachbarschaftsbeziehungen nicht statisch festgelegt, sondern können selbst entweder diskrete Ereignisse nach sich ziehen oder von einem diskreten Ereignis ausgelöst werden. Dazu betrachten wir im Modell Bedingungen der Form:

$$b_d = 1 \Rightarrow \|d_{i_1, i_2}\| < D_1 \quad (4.22a)$$

$$b_d = 1 \Rightarrow \|d_{i_1, i_2}\| > D_2 \quad (4.22b)$$

$$\|d_{i_1, i_2}\| < D_1 \Rightarrow b_d = 1 \quad (4.22c)$$

$$\|d_{i_1, i_2}\| > D_2 \Rightarrow b_d = 1 \quad (4.22d)$$

Im Fall der Gleichungen (4.22a) oder (4.22b) kann b_d beispielsweise eine diskrete Steuervariable sein, mit der das Einnehmen einer Formation erzwungen wird. Die Gleichungen (4.22c) und (4.22d) beschreiben Ereignisse, die als Folge einer Abstandsbeziehung auftreten. Ein Beispiel dafür ist das Eintreten in den Kommunikationsbereich oder das Erreichen der Sensorreichweite eines beteiligten Fahrzeugs.

Die beschriebenen paarweisen Bedingungen stellen einerseits ein zentrales Charakteristikum kooperativer Mobilität dar, bringen aber andererseits auch große Herausforderungen in das zu lösende Optimierungsproblem ein; die Zustandsvariablen des Differentialgleichungssystems werden aneinander gekoppelt, der diskrete Problemanteil erhöht sich und die Konvexität des Steuer-Zustands-Raumes kann verloren gehen.

Es wird dazu nun kurz auf die Modellierung zweier wichtiger Aspekte eingegangen, die in den folgenden Abschnitten und im Kapitel 5 aufgegriffen werden.

Anwendungsbeispiel: Funk-Konnektivität zwischen Fahrzeugen. Ausgangspunkt ist die bislang in der Literatur wenig untersuchte Frage: „Wie sieht ein optimaler Bewegungsablauf kooperierender Fahrzeuge aus, wenn dabei eine sichere Funkkommunikation garantiert werden soll?“

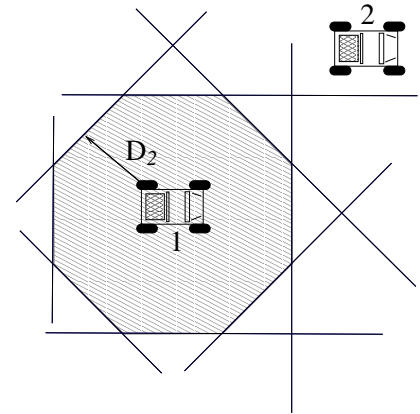


Abbildung 4.5: Approximation der euklidischen Norm durch $n_d = 8$ Geraden.

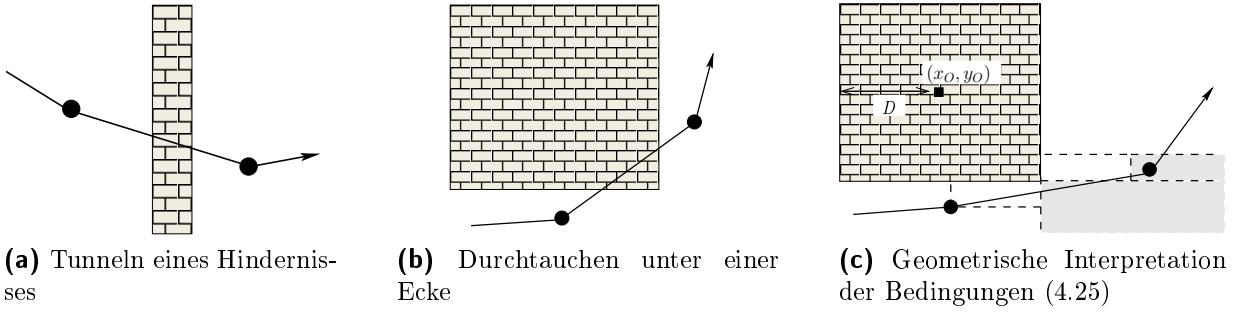


Abbildung 4.6: Trotz korrekt berechneter Wegpunkte kann eine Trajektorie unzulässig sein. Weitere Bedingungen (z.B. Gl. (4.25)) sind nötig.

Für die Funkübertragung zeigt sich eine detaillierte Modellierung als sehr komplex, insbesondere in strukturierten Umgebungen. Zu den wichtigsten physikalischen Problemen der Funkkommunikation bietet beispielsweise [Lan99] eine Übersicht.

Um in das Mehrfahrzeugmodell Beschränkungen resultierend aus Kommunikationsforderungen aufzunehmen, muss die Sende-Empfangs-Charakteristik drahtloser Informationsübertragung berücksichtigt werden. Für jedes Fahrzeug wird dazu eine Funktion $C_{i_1}(\cdot, \cdot)$ betrachtet, welche die Sende-Signalstärke von Fahrzeug i_1 in Abhängigkeit seiner Position und einer Empfängerposition (x^{i_2}, y^{i_2}) beschreibt. Diese kann im Allgemeinen sowohl durch einfache Distanzannahmen eines Ausbreitungsmodells im freien Raum oder durch ein detailliertes Raytracing-Modell mit Reflektionen und slow- bzw. fast-fading Effekten motiviert sein. Ebenso sind stochastische Modelle möglich.

Auf der stetigen Funktion $C_{i_1}(\cdot, \cdot)$ basierend wird für ein Fahrzeugpaar (i_1, i_2) und eine nötige Signalstärke c_{i_2} eine binäre Variable $b_{i_1, i_2} \in \{0, 1\}$ eingeführt,

$$[b_{i_1, i_2}(t) = 1] \Leftrightarrow [C_{i_1}((x^{i_1}(t), y^{i_1}(t)), (x^{i_2}(t), y^{i_2}(t))) \geq c_{i_2}] . \quad (4.23)$$

Wir gehen davon aus, dass die Bedingung in (4.23) für eine feste Position (x^{i_1}, y^{i_1}) und ein festes c_{i_2} eine konvexe Menge definiert und sich (4.23) in der Form

$$[b_{i_1, i_2} = 1] \Leftrightarrow [\|\mathbf{d}_{i_1, i_2}\|_{\mathbf{M}} \geq c_{i_2}] \quad (4.24)$$

schreiben lässt. Dabei meint $\|\mathbf{d}\|_{\mathbf{M}}$ eine der Normen $\|\mathbf{M}\mathbf{d}\|_1$, $\mathbf{d}^T \mathbf{M} \mathbf{d}$ oder $\|\mathbf{M}\mathbf{d}\|_{\infty}$ mit positiv definiter Matrix \mathbf{M} .

Anwendungsbeispiel: Kollisions- und Hindernisvermeidung. Agieren Fahrzeuge auf relativ engem Raum, besitzt Kollisionsvermeidung eine wichtige Bedeutung. Die Optimallösung kooperativen Bewegungsverhaltens wird davon maßgeblich beeinflusst. Hindernisse bilden einen Spezialfall der Kollisionsvermeidung, indem sie als ruhende Fahrzeuge betrachtet werden.

Wegen seiner Sicherheitsrelevanz weisen wir hier auf eine Problematik hin, die sich ergibt, wenn man sich in der approximativen Berechnung nur auf einzelne Wegpunkte stützt. Abbildung 4.6 zeigt Situationen, in denen eine Trajektorie trotz korrekt berechneter Wegpunkte unzulässig ist. Liegt eine Modellierung gemäß Gleichung (4.21) vor, so können Situationen der Gestalt in Abb. 4.6a in der Modellbildung über zusätzliche Bedingungen

an den zeitlichen Verlauf der binären Variablen b_l berücksichtigt werden. Ein Tunneln an den Ecken eines rechteckigen Hindernisses (Abb. 4.6b) verhindern die Bedingungen

$$x^{i_1}(k) + x^{i_1}(k+1) - b_v(k) (2 \cdot (x_O + D) + M_y) > -M_x \quad (4.25a)$$

$$y^{i_1}(k) + y^{i_1}(k+1) - b_v(k) \cdot M_y < 2(y_O - r) + M_y \quad (4.25b)$$

mit $M_x \geq 2 \cdot \max\{x^{i_1}\}$, $M_y \geq 2 \cdot \max\{y^{i_1}\}$ und (x_O, y_O) als zentralem Punkt im Hindernis. Dabei wird diese kritische Situation ebenfalls über eine Bedingung an den zeitlichen Verlauf der binären Variablen b_l identifiziert, so dass $b_v = 1$ genau dann gilt, wenn ein Fahrzeugpfad die Ecke v passiert.

Weitere Bedingungen an die binären Variablen b_v lassen sich angeben. Mit

$$\forall k \forall v_1 \neq v_2 : b_{v_1}(k) + b_{v_2}(k) \leq 1 \quad (4.26)$$

wird so etwa eine zur Optimierung verbesserte Beschreibung erzielt, die besagt, dass von einem Fahrzeug pro Zeitschritt keine zwei Ecken gleichzeitig passiert werden können.

In der Literatur finden sich dazu alternativ Ansätze, die sukzessive das Zeitgitter verfeinern [ED04], Hindernisse in Übergröße betrachten [BF06] oder interpolierende Trajektorienpunkte zwischen den zu optimierenden Wegpunkten in das Modell mit aufnehmen [SFH06]. Demgegenüber erhält der Ansatz der Gl. (4.25) sowohl die Effizienz von MILP und benötigt lediglich eine zusätzliche binäre Variable pro Ecke und Zeitschritt bei geringem Genauigkeitsverlust.

4.3.2 Veränderliche Graphenstrukturen:

Nachdem der vorige Abschnitt die Frage des Eintretens und Verlassens einer Nachbarschaftsbeziehung zweier Fahrzeuge betrachtet hat, liegt das Augenmerk nun auf der Beschreibung einer variablen Graphenstruktur, wie sie etwa bei wechselnden Formationen oder in der Netzwerktopologie der Drahtloskommunikation vorkommt.

Allgemein kann man Bedingungen für einen Informationsaustausch angeben, der in erster Linie durch ein permanent bestehendes, stabiles Konnektivitätsnetzwerk gesichert wird. Dies umfasst, neben Forderungen an die Beständigkeit der einzelnen Funkverbindungen, Beschränkungen an die Netzwerktopologie und gegebenenfalls an die Häufigkeit von Topologiewechseln.

Die zeitabhängige, quadratische Matrix $\mathcal{A}(t) = (b_{i_1, i_2}(t)) \in \{0, 1\}^{n_v \times n_v}$ beschreibe alle möglichen Übertragungen von i_1 zu i_2 (vgl. dazu den Ausdruck (4.24)) und definiere somit eine Nachbarschaftsmatrix der Kommunikations-Netzwerktopologie:

$$b_{i_1, i_2}(t) := \begin{cases} 1 & \text{falls Fahrzeug } i_1 \text{ und } i_2 \text{ verbunden,} \\ 0 & \text{sonst.} \end{cases} \quad (4.27)$$

In den meisten Anwendungen mit ausgeprägter Kooperation wird eine wechselseitige Kommunikation gewünscht und es werden lediglich symmetrische Matrizen $\mathcal{A} = \mathcal{A}^T$ zugelassen. Ein Wechsel in der Netzwerkstruktur soll nur zu bestimmten Schaltzeitpunkten stattfinden, so dass

$$\mathcal{A}(k) := \mathcal{A}(t) \text{ für } t_{k-1} \leq t \leq t_k, \quad k = 1, \dots, n_c \quad (4.28)$$

definiert werden kann. Falls das Zeitgitter $(t_k)_{1, \dots, n_c}$ variabel ist, erlauben weitere Beschränkungen der Art $t_{k-1} - t_k \geq \Delta_{t, \min}$ Beständigkeitsforderungen der Netzwerktopologie zu formulieren.

Mit $\forall t \in [t_0, t_f) : \mathcal{A}(t) \in \mathcal{C}$ wird berücksichtigt, dass \mathcal{A} aus einer endlichen Menge zulässiger Netzwerktopologien \mathcal{C} stammt. Die Menge der Elemente in \mathcal{C} wird mit logischen Ausdrücken beschrieben, beziehungsweise entsprechenden algebraischen Formulierungen an binäre Variablen. Im Hinblick auf Ausfallsicherheit kann dazu beispielsweise eine gewünschte δ -Konnektivität formuliert werden und für Quality-of-service-Untersuchungen kann eine k -hop-Konnektivität betrachtet werden.

Eine zentrale Eigenschaft der Elemente in \mathcal{C} ist dabei die Verbundenheit des Netzwerkes. Als Maß dafür wird (z.B. auch in [ZP05, FTEBJ06, MZKP08]) der „Fiedler-Eigenwert“ [Fie73] („algebraische Konnektivität“) der algebraischen Graphentheorie verwendet. Es ist dies der zweitgrößte Eigenwert λ_2 der *Laplace-Matrix*

$$\mathcal{A}_L := \text{diag}\left(\sum_{\substack{i_2=1 \\ i_2 \neq i_1}}^{n_v} b_{i_1, i_2}, \dots, \sum_{\substack{i_2=1 \\ i_2 \neq i_{n_v}}}^{n_v} b_{i_{n_v}, i_2}\right) - \mathcal{A},$$

für den insbesondere gilt: $[\lambda_2 > 0 \Leftrightarrow \text{Graph ist zusammenhängend}]$.

Diese Formulierung kann entweder selbst in eine nichtlineare Problemformulierung eingehen oder dazu verwendet werden, Elemente der Menge \mathcal{C} zu bestimmen, um sie anschließend mit Hilfe linearer Ausdrücke zu umschreiben.

Verwandte Ansätze in der Literatur finden sich unter dem weiten Feld der „topology control“, wobei man hier wiederum zwischen probabilistischen und algorithmischen Ansätzen unterscheidet.

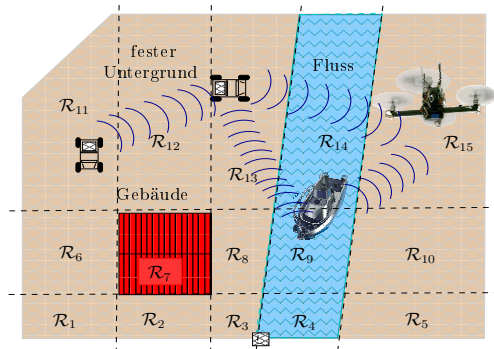
Als algorithmischer Ansatz (z.B. in [LH04, SM04]) werden Funktionen konstruiert, welche die Robustheit lokaler Konnektivität messen, so dass sich daraus hinreichende Bedingungen für die globale Verbundenheit konstruieren lassen. In [BR04] wurden dynamische Netzwerke mit einfachen Distanz-Beschränkungen untersucht und ein Algorithmus vorgestellt, welcher die Netzwerktopologie durch Verschieben weniger Knoten verbessert. Allerdings geschehen solche Untersuchungen in der Regel, ohne spezifische physikalische Bewegungsfähigkeiten zu berücksichtigen.

In [NSBJ06] wurden Untersuchungen mit Einheitskreisgraphen und einer vereinfachten Dynamik zweiter Ordnung gemacht, um Beschränkungen an die kontinuierlichen Steuervariablen zu generieren. Eine detaillierte graphentheoretische und algebraische Formulierung zur Modellierung von k -hop-Konnektivitätsbeschränkungen wird in [ZP05] präsentiert. Weiterhin wird dort ein zentraler Rahmen zur Steuerung der Struktur dynamischer Graphen vorgestellt. Der verteilte Ansatz in [MZKP08] garantiert Konnektivität und verwendet einen marktbasierten Austausch über zu löschende Verbindungen. Grundlage dafür ist eine lokale Approximation der Netzwerküte.

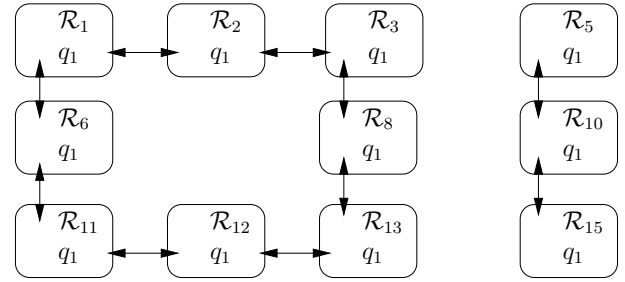
Die numerische Optimalsteuerung eines Mehrfahrzeugsystems unter den Beschränkungen dynamischer Graphenstrukturen der Funk-Netzwerkcommunication wurde in der Literatur bislang noch kaum untersucht.

Graphenstruktur zur Beschreibung eines Operationsgebietes: Eine strukturierte Umgebung, die verschiedene Bereiche wie Hindernisse und unterschiedliche physikalische Charakteristiken besitzt – zum Beispiel Wasseroberflächen, verschiedene Bodenbeschaffenheiten und freien Luftraum – verlangt nach einer fahrzeugspezifischen Beschreibung.

Gemäß den Fortbewegungsfähigkeiten und den Gegebenheiten des Gebietes gilt es beispielsweise, optimale Fahrzeugsteuerungen kooperativer Erkundungsmissionen zu bestim-



(a) Partitionierung eines strukturierten Gebietes, welches ein Gebäude (\mathcal{R}_7) und einen Fluss ($\mathcal{R}_4 \cup \mathcal{R}_9 \cup \mathcal{R}_{14}$) enthält.



(b) Graph zusammenhängender zulässiger Teilbereiche für ein Bodenfahrzeug in Abb. 4.7a.

Abbildung 4.7: Fahrzeugspezifische Beschreibung des Umfeldes durch eine Graphenstruktur.

men. Dazu werde verlangt, dass eine Menge von Teilgebieten während der Mission von mindestens einem Fahrzeug passiert wird.

Die zweidimensionale Projektion $\mathcal{R} \subset \mathbb{R}^2$ des Gebietes (vgl. Abb. 4.7a), das sich in eine Menge konvexer Teilregionen partitionieren lässt ($\mathcal{R} = \bigcup_{l=1}^{n_R} \mathcal{R}_l$) bildet den Ausgangspunkt der Beschreibung. Jedem Teilbereich \mathcal{R}_l ist ein diskreter Parameter $q_l \in Q$ zugeordnet, der ein konkretes physikalisches Merkmal repräsentiert, das den fahrzeugspezifischen Bewegungsmodus für den Teilbereich bestimmt. Beispielsweise werden die Zustände {fester_Boden, Wasser, Gebäude} unterschieden und durch $q_l \in Q = \{1, 2, 3\}$. Die zulässigen, fahrzeugspezifischen Sequenzen von Bereichen werden als gerichteter Graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ (vgl. Abb. 4.7b) dargestellt, mit $\mathcal{V}_i \subset \{\mathcal{R}_l \mid l = 1, \dots, n_R\}$ und $\mathcal{E}_i \subset \mathcal{V}_i \times \mathcal{V}_i$. Handelt es sich bei den Teilregionen \mathcal{R}_l um Polygone, so lässt sich im Fall linearer zeitdiskreter Beschreibung ein PWA-Modell direkt angeben.

4.3.3 Spezialfall: Mobilität im dichten drahtlosen Sensornetzwerk

Drahtlos-Sensor-Netzwerke (Wireless Sensor Networks - WSN) sehen sich wachsenden Anwendungen gegenüber, etwa in der Umweltbeobachtung oder beim Brandschutz. Typischerweise kooperieren dabei mobile Knoten und stationäre Sensoren zur Erfassung räumlich verteilten Daten.

Obwohl einzelne stationäre Sensorknoten häufig nur eine labile Zuverlässigkeit aufweisen, erwartet man, dass ein WSN verlässliche Dienste bereitstellt. Dies wird vor allem durch eine Redundanz von Sensorknoten gesichert, was zu einem – im Verhältnis zur Kommunikationsreichweite – dicht liegenden Netz von Sensorknoten führt.

In [GT02] wurde gezeigt, dass Knotenmobilität die Kapazitäten eines Ad-hoc-Netzwerkes signifikant erhöhen kann, falls mobile Knoten Nachrichten in Richtung Ziel transportieren, anstelle diese über eine Multi-hop-Kommunikation der Sensorknoten weiterzureichen. Für nicht zeitkritische Daten ist ein solcher Übertragungsweg also vorteilhaft.

Als interdisziplinäre Fragestellung wurde im Umfeld der vorliegenden Arbeit untersucht, wie die Mobilität einzelner Knoten die Langlebigkeit und Zuverlässigkeit eines WSN begünstigen kann. Dazu wird eine eigene Methode der Pfad- und Trajektorienplanung kooperierender mobiler Knoten zur Unterstützung im Transport nicht zeitkritischer Daten

vorgestellt. Verwandte Aspekte der Fragestellung, wie etwa die effiziente Erfassung von Daten und die kartenbasierte Darstellung wird in den Publikationen [KSA⁺10, KRA⁺10] detailliert erläutert.

In der Literatur gibt es verschiedene mobilitätsgestützte Ansätze zur Datensammlung. Wir nennen verkürzt einige dieser Ansätze und verweisen auf die ausführliche Diskussion in [KSA⁺10, KRA⁺10] sowie auf den Überblicksartikel [WHW07].

- Daten-Muli (*Data Mule*) ist die Bezeichnung für (langsame) mobile Knoten, die jeweils Daten aufnehmen, wenn sie sich nahe eines Sensorknotens bewegen.
- Nachrichtenfähren (*Message Ferrying*) sind ähnlich den Daten-Mulis, allerdings für mobile ad-hoc-Netzwerke entworfen, um hauptsächlich Netzwerkpartitionierungen auszugleichen.
- Mobile Senken und „Unterstützungsknoten“ (assist nodes) umfassen Ansätze, die ähnlich denen von Daten-Mulis sind, allerdings für Delay-kritische Anwendungen. Sie basieren auf einer mobilen Senke oder Relay-Knoten.

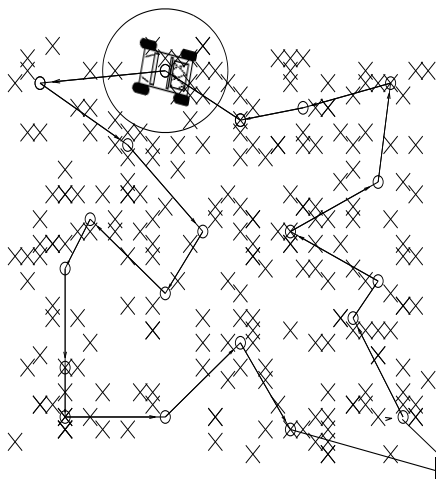


Abbildung 4.8: Ein Fahrzeug kommuniziert an Stopppositionen (o) mit den Sensorknoten (x) und transportiert die Daten zur Senke (■).

Demgegenüber wird eine eigene, optimierungsbasierte Methode vorgestellt, die auf allgemeine, insbesondere räumlich dichte WSN und kooperierende Fahrzeuge angewendet werden kann. Dabei berücksichtigt die Planung der kooperativen Mobilität eine gesicherte Kommunikationsmöglichkeit zu allen Sensorknoten und einen koordinierten Bewegungsablauf der beteiligten Fahrzeuge.

Betrachtet wird ein Szenario (vgl. Abb. 4.8) mit einer stationären Netzwerksenke als Benutzer-schnittstelle, einer Vielzahl dicht liegender Sensorknoten und wenigen mobilen Unterstützungsknoten. Die mobilen Knoten besitzen funktionale Fähigkeiten und stehen beispielsweise für mobile Roboter oder unbemannte Luftfahrzeuge. Sie seien steuer- und beobachtbar und die Positionen aller Sensorknoten seien hinreichend genau bekannt. Es wird angenommen, dass die Kommunikation zwischen Sensorknoten und einem mobilen Knoten jeweils garantiert werden kann, wenn sich dieser für eine feste Zeitspanne an einer festen Position innerhalb einer spezifischen Umgebung des Sensorknotens befindet.

Ziel ist es, optimale Bahnen zur kooperativen Datensammlung zu planen, so dass möglichst wenig redundanter Funkverkehr entsteht. Eine Bahn ist optimal, wenn sie durch eine minimale Zahl von Kommunikations-Haltepunkten definiert wird, wenn sich die Bereiche sicherer Konnektivität möglichst wenig überlappen und wenn die Gesamtlänge der Tour möglichst kurz ist.

Ziel ist es, optimale Bahnen zur kooperativen Datensammlung zu planen, so dass möglichst wenig redundanter Funkverkehr entsteht. Eine Bahn ist optimal, wenn sie durch eine minimale Zahl von Kommunikations-Haltepunkten definiert wird, wenn sich die Bereiche sicherer Konnektivität möglichst wenig überlappen und wenn die Gesamtlänge der Tour möglichst kurz ist.

Aufgrund der hohen Zahl von Konnektivitätsbeziehungen zwischen mobilen Knoten und Sensorknoten ist die Fragestellung durch einen kombinatorischen Problemanteil dominiert. Die Annahme, dass eine Kommunikation nur an Stopppunkten stattfindet, erlaubt es jedoch, das Problem der Bestimmung dieser Positionen von der Trajektorienplanung abzu-

koppeln. Dadurch entsteht ein spezielles *Problem mehrerer Handlungsreisender mit Nachbarschaft* (s. a. S. 71, multiple travelling salesmen problem with neighborhood - multi-TSPN), für das folgende Teilprobleme getrennt betrachtet werden: (1) Das Bestimmen einer reduzierten Menge an Kommunikations-Haltepunkten, (2) die Reduzierung von redundanten Kommunikation und (3) die Planung der kooperativen Fahrzeugtrajektorien.

1) Bestimmung einer minimalen Zahl an Kommunikations-Haltepunkten: Betrachtet werden nur kreisrunde Konnektivitätsgebiete mit Radius R . Der Ansatz lässt sich aber einfach auf beliebige konvexe Konnektivitätsgebiete erweitern.

Um die Lösung eines hochdimensionalen, gemischt-ganzzahligen NLP zu umgehen, wird für gegebene Sensorpositionen $\mathbb{P}_S := \{(\xi^j, \eta^j) \mid j = 1, \dots, N_S\}$ ein approximativer Algorithmus (Alg. 2) vorgeschlagen.

Algorithmus 2: Reduzierung und optimale Positionierung von Haltepunkten

- 1: Initialisiere Menge an Haltepunkten $\mathbb{P}_C := \{(x^j, y^j) \mid j = 1, \dots, N_C\}$, die Konnektivität mit allen Sensorknoten garantiert.
 - 2: **while** (Zahl der Positionen N_C nimmt ab) **do**
 - 3: Bestimme zu jedem Haltepunkt $(x^j, y^j) \in \mathbb{P}_H$ die Indizes aller Sensorknoten, welche innerhalb der Voronoi-Zelle von (x^j, y^j) liegen:

$$I_j^{CS} = \left\{ i \mid j = \arg \min_{k=1, \dots, N_C} \{ \|(x^k, y^k) - (\xi^i, \eta^i)\| \} \right\}$$
 - 4: Bestimme zu jedem (x^j, y^j) die Indexmenge benachbarter Haltepunkte:

$$I_j^{CC} = \{ i \mid \|(x^j, y^j) - (x^i, y^i)\| \leq k_1 \cdot R \}$$
 - 5: Minimiere die Straffunktion $\varphi((x^1, y^1), \dots, (x^{N_C}, y^{N_C}))$ unter Beibehaltung von Konnektivitätsbedingungen.
 - 6: Prüfe auf redundante Überdeckungen von Konnektivitätsgebieten und lösche ggf. Haltepunkte aus \mathbb{P}_C .
 - 7: **if** (\mathbb{P}_C konnte nicht reduziert werden) **then**
 - 8: Maximiere (approximativ) die Straffunktion $\varphi((x^1, y^1), \dots, (x^{N_C}, y^{N_C}))$ unter Beibehaltung von Konnektivitätsbedingungen.
 - 9: Prüfe auf redundante Überdeckungen von Konnektivitätsgebieten und lösche ggf. Haltepunkte aus \mathbb{P}_C .
 - 10: **end if**
 - 11: **end while**
 - 12: Gebe $\mathbb{P}_C^* := \mathbb{P}_C = \{(x^j, y^j) \mid j = 1, \dots, N_C^*\}$ zurück
-

Die Minimierung der nichtlinearen Straffunktion

$$\begin{aligned}
 & \min_{\substack{(x^j, y^j) \\ j=1, \dots, N_C}} \varphi((x^1, y^1), \dots, (x^{N_C}, y^{N_C})) \\
 &= \min_{\substack{(x^j, y^j) \\ j=1, \dots, N_C}} \sum_{i \in I_j^{CS}} \log(\|(x^i, y^i) - (x^j, y^j)\| + 1)
 \end{aligned} \tag{4.29}$$

unter Einhaltung der Nebenbedingungen $\forall k \in I_j^{CS} : \|(x^j, y^j) - (\xi^k, \eta^k)\| \leq R$ bewirkt, dass nahe beieinander liegende Haltepunkte sich weiter aufeinander zu bewegen und idealerweise zusammenfallen.

Die Effizienz der Lösung von (4.29) hängt maßgeblich von der Anzahl betrachteter Sensorpositionen N_S und Stopppositionen N_C ab. Es ist also bereits bei der ersten Initialisierung der Menge \mathbb{P}_C darauf zu achten, dass diese möglichst klein ist. Ferner lässt sich der Algorithmus über Nachbarschaftsparameter k_1 justieren. Durch ein zwischenzeitliches, approximatives Maximieren der Straffunktion entsteht eine Veränderung der Menge benachbarter Knoten. Dies zeigt als hilfreich, lokale Minima von N_c erneut zu verbessern. Eine Evaluierung und Ergebnisse zu Algorithmus 2 finden sich in Abschnitt 5.2.1.

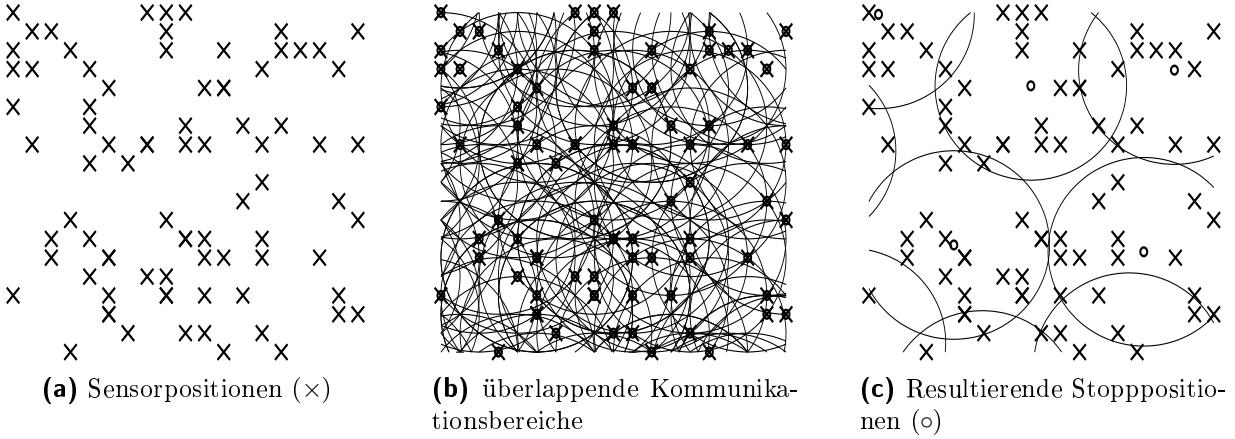


Abbildung 4.9: Berechnung optimaler Haltepositionen zur Kommunikation mit Sensoren.

2) Reduzierung redundanter Kommunikation: Eine redundante Konnektivitätsgarantie liegt vor, wenn eine Sensorposition im Kommunikationsbereich zweier Haltepunkte liegt. Eine Reduktion solcher Redundanzen für N_C^* Haltepunkte kann durch das Lösen eines gemischt-ganzzahligen NLP berechnet werden:

$$\min \sum_{(j,k) \in J^{CS}} b_{jk}$$

unter den Nebenbedingungen

$$\forall (j,k) \in J^{CS} \quad b_{jk} = 1 \Leftrightarrow \|(x^j, y^j) - (\xi^k, \eta^k)\| < R \quad (4.30a)$$

$$\forall k \in \mathbb{P}_S : \sum_{j=1}^{N_C^*} b_{jk} \geq 1 \quad (4.30b)$$

für $J^{CS} = \{(j,k) \in \{1, \dots, N_C^*\} \times \{1, \dots, N_S\} \mid \|(x^j, y^j) - (\xi^k, \eta^k)\| < k_2 \cdot R\}$ und binäre Variablen $b_{jk} \in \{0, 1\}$.

Aus Effizienzgründen schlagen wir für dieses unter Umständen hochdimensionale Problem ein iteratives Optimieren auf lokalen Nachbarschaften $J^{CS} = \{(j,k) \in \{1, \dots, N_C^*\} \times \{1, \dots, N_S\} \mid \|(x^j, y^j) - (\xi^k, \eta^k)\| < k_2 \cdot R\}$ und das Verwenden linearer Approximation für die Norm (4.30a) gemäß Gleichung (4.21) (S. 61) vor. Verglichen mit dem nichtlinearen Problem hat das MILP eine höhere diskrete Struktur, liefert dafür aber stets ein globales Minimum, ist wesentlich effizienter lösbar und erlaubt es, Zeitbeschränkungen zur suboptimalen Lösung im Algorithmus einzubinden.

3) Planung der kooperativen Fahrzeugtrajektorien: Unter der Annahme, dass ein mobiler Knoten zur Kommunikation seine Fortbewegung unterbrechen muss, können Kosten, um von Haltepunkt (x_{j_1}, y_{j_1}) nach (x_{j_2}, y_{j_2}) zu kommen, konstant und unabhängig von der Reihenfolge aller vorhergehend und nachfolgend besuchten Punkte angenommen werden.

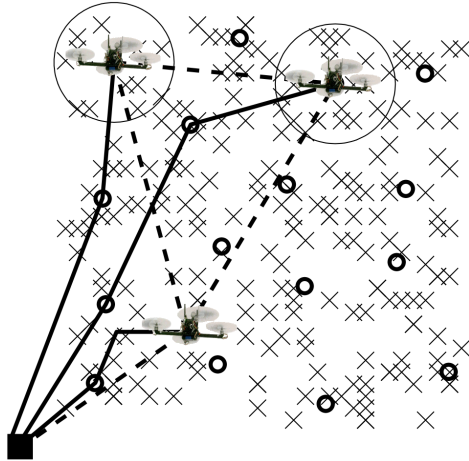


Abbildung 4.10: Fahrzeuge können mit hoher Kommunikationsreichweite Informationen austauschen.

Somit stellt das zu lösende Pfadplanungsproblem im Fall, dass nur ein Fahrzeug betrachtet wird, ein symmetrisches Handlungsreisendenproblem dar.

Werden mehrere mobile Knoten berücksichtigt, so lässt sich deren Grad an Kooperation durch koordinierte Bewegungen erheblich steigern. Die Grundidee besteht dabei darin, die Fahrzeuge so an den Kommunikationshaltepunkten zu positionieren, dass diese auch untereinander und ggf. mit der Netzwerkesenke kommunizieren können (vgl. Abb. 4.10). Um Wege zu optimieren, welche diese Eigenschaften garantieren, ist es erforderlich, den zeitlichen Verlauf der Fahrzeugbewegung zu berücksichtigen. Entscheidend ist nicht nur die Aufteilung und Reihenfolge der anzufahrenden Kommunikationshaltepunkte, sondern auch, dass die Fahrzeuge zu *konkreten Zeit-*

punkten eine zulässige Formation bilden. Weitere Details und exemplarische Ergebnisse werden in Abschnitt 5.2.1 vorgestellt.

4.4 Möglichkeiten der Auskopplung von Teilproblemen

Die Motivation der Auskopplung liegt in der hohen Berechnungskomplexität, insbesondere des modellprädiktiven Gesamtproblems. Neben dem Ansatz, einzelne Variablen der Optimalsteuerung auf Basis einer heuristischen Methode festzulegen, steht der Gedanke, die Berechnung möglichst auf teilnehmende Fahrzeuge zu verteilen. Dazu wiederum existieren zwei grundsätzlich unterschiedliche Vorgehensweisen.

Die erste Möglichkeit ist eine *verteilte Berechnung* des gemischt-ganzzahligen Problems. Dabei wird jeweils das Optimierungsproblem mit einer Auswahl an Nebenbedingungen den Fahrzeugen zugeordnet, so dass diese darauf basierende suboptimale Steuerungen berechnen und dabei die gefundenen Optima über eine Kommunikationsinfrastruktur austauschen. Ein ähnlicher Ansatz wird zum Beispiel auch in [BF06, NB07, DS09b] vorgestellt.

Ein alternativer Ansatz besteht in dem *modellbasierten Entwurf einer mehrschichtigen Steuer- und Regelungsarchitektur*, der einzelne Teile des Gesamtmodells in eine dezentrale Regelung auslagert. So kann beispielsweise eine Aufgabenzuweisung und grobe Pfadplanung zentral stattfinden, während das Anfahren von Wegpunkten und eine Kollisionsvermeidung dezentral, lokal geregelt wird. Beispiele für dieses Vorgehen finden sich unter anderem in [KFB⁺06, KRSH06, KBF⁺08].

Zwei typische Teilprobleme, die für die Entwicklung hierarchischer Reglerarchitekturen oftmals ausgekoppelt werden, werden in den nächsten beiden Abschnitten diskutiert.

4.4.1 Grundproblem Aufgabenzuweisung

Ein weit verbreiteter Ansatz in vielen realen Anwendungen ist die Abkopplung der Aufgabenzuweisung vom Gesamtproblem. Die Verteilung konkreter Aufgaben geschieht ohne genaue Beachtung einer physikalischen Bewegungsdynamik. Man hofft, damit den diskreten Problemanteil zu verkleinern oder sogar ganz auszulagern. Für Probleme, bei denen Bewegung entscheidenden Einfluss auf die Qualität einer kooperativen Aufgabenerfüllung hat, bedeutet dies aber unter Umständen einen erheblichen Optimalitätsverlust. Beispiele dafür stellen die in der Einleitung genannten verhaltensbasierten, marktbasieren, dezentralen Reglerstrukturen dar.

Die im Rahmen der Arbeit entwickelte optimierungsbasierte Methode dient einerseits dem Anwender als Referenzlösung, um den Optimalitätsverlust, welcher durch eine rein heuristische Aufgabenzuweisung erfolgt, zu messen, und bietet andererseits auch die Möglichkeit, selbst zentrale Komponente einer der genannten Reglerstrukturen zu sein.

Um die Aufgabenzuweisung für den speziellen Anwendungsfall zu erleichtern und gegebenenfalls an eine heuristische Reglerstruktur auszulagern, schlagen wir folgende Kriterien vor, welche hilfreich in die Planung eingebracht werden können:

1. Falls nicht alle der zuzuweisenden Aufgaben in gleicher Weise von der Mobilität abhängen, können für einige Teilaufgaben gegebenenfalls feste Kostenschätzungen berechnet werden und für eine hinreichend gute Zuweisung herangezogen werden.
2. Können zu den Teilaufgaben minimale und maximale Kosten abgeschätzt werden, so lässt dies eine Vorauswahl und ein reduziertes diskret-kontinuierliches Problem zur Lösungsfindung zu.
3. Besteht das System aus verschiedenartigen Fahrzeugen, die eine fahrzeugspezifische Priorisierung der Aufgaben erlaubt, kann diese ebenfalls die Zuweisung unterstützen.
4. Enthält das betrachtete Optimierungsproblem Lösungen, die unter allen Umständen identisch bewertet werden, so lässt sich die Effizienz eines Optimierungsverfahrens unter Umständen deutlich steigern, wenn man durch zusätzliche Nebenbedingungen eine Modellverbesserung vornimmt, welche nur noch eine möglichst eindeutige Lösung im Suchraum zulässt (Anwendung siehe Abschnitt 5.1.1).

4.4.2 Handlungsreisendenprobleme

Sowohl Handlungsreisendenprobleme (*travelling salesman problem* – TSP) als auch *vehicle-routing-Probleme* (VRP) [Sol87] stellen für eine Vielzahl an Erkundungs- und Beobachtungsaufgaben ein Kernproblem dar, das zur approximativen Zuweisung von Wegpunkt-reihenfolgen verwendet wird (Bsp. [GHAL05]). Auch hier besteht die Motivation wieder darin, den diskreten Problemanteil im Gesamtproblem auszukoppeln und die eigentliche Trajektorienplanung auf ein rein kontinuierliches Problem zu reduzieren.

Für die Fragestellungen kooperativer Fahrzeugsysteme sind vor allem Probleme mehrerer Handlungsreisender (*multiple TSP* – mTSP oder auch k-TSP) bedeutsam. Diese lassen sich auf einfache TSP transformieren (zum Beispiel [JV88]). TSP und VRP sind ferner für viele Anwendungen des Operations Research von großer Wichtigkeit und wurden deshalb in vielerlei Ausprägungen wissenschaftlich untersucht; so zum Beispiel symmetrische und asymmetrische TSP, mTSP und VRP in min-max-Formulierung, unter Berücksichtigung

beschränkter Depots, unter Einbeziehung von Zeitfenstern, und so weiter. Ausführlicher wurde die Historie dieser Probleme in [KRA⁺10] beschrieben. Wir konzentrieren uns hier auf die Betrachtung zweier bedeutsamer Spezialfälle: das *Handlungsreisendenproblem mit Nachbarschaft* und das *Problem mehrerer motorisierter Handlungsreisender*.

Handlungsreisendenproblem mit Nachbarschaft: Die Anwendungen von Mehrfahrzeugsystemen – etwa in der kooperativen Erkundung oder in der Datensammlung in WSN (Abschnitt 4.3.3) – verlangen häufig nicht, dass einzelne Positionen exakt angefahren werden, sondern es genügt, eine Umgebung („Nachbarschaft“) dieser Wegpunkte zu besuchen. Das TSP, welches dies berücksichtigt, ist als *Handlungsreisendenproblem mit Nachbarschaft* (TSPN – siehe dazu z.B. [YOS07] und Referenzen dort) bekannt. Die Planung eines kürzesten Pfades für ein Einzelfahrzeug ist bereits \mathcal{NP} -schwer. Dies ist auch der Grund, weshalb in den vergangenen Jahren zahlreiche Approximationsalgorithmen für TSPN-Probleme entwickelt wurden, die unter schwachen Voraussetzungen gute Ergebnisse erzielen, z.B. [Mit07, EFS06].

Zur Bestimmung eines kooperativen Fahrzeugverhaltens sind mTSP(N) und VRP nur bedingt einsetzbar, da sie keinerlei zeitliche Systementwicklung berücksichtigen und somit aus der Lösung noch kein koordinierter Bewegungsablauf folgt.

Probleme mehrerer motorisierter Handlungsreisender: In Anwendung auf kooperative Erkundungsaufgaben (Abschnitt 5.1) und in mobilitätsgestützten WSN (Abschnitt 5.2) werden im folgenden Kapitel repräsentative Probleme vorgestellt und untersucht, in denen ein mTSP(N) mit zeitlicher Entwicklung die Grundlage der Modellierung bildet. Geschieht die Modellierung auf Basis einer physikalischen Fahrdynamik, ist das Problem auch als *motorisiertes Handlungsreisendenproblem* bekannt. Es stellt ein Benchmarkproblem der diskret-kontinuierlichen Optimalsteuerung und der kooperativen Mobilität dar. Es wurde beispielsweise in [GvS02, Käm06, SFB08] untersucht.

Zusammenfassung des Kapitels

Ein neues Modellierungskonzept für das breite Spektrum kooperativer, mobiler Trajektorien- und Aufgabenplanung wurde vorgestellt. Dabei werden verschiedene Abstraktionsebenen zugelassen und die Problemformulierungen einer Optimierung zugänglich gemacht.

Insbesondere wurden Teilmodelle für Kommunikationsanforderungen, für veränderbare Graphenstrukturen und für die Beschreibung des Operationsgebietes hervorgehoben. Bestehende MILP-Formulierungen wurden in das Konzept integriert und weiterentwickelt, beispielsweise in der Hindernisbehandlung, der Betrachtung von minimalen Kurvenradien und distanzabhängigen Beschränkungen. Insbesondere auf die Vorzüge der linearen Beschreibung von Konnektivitätsanforderungen in bislang noch nicht untersuchten Fragestellungen koordinierter Bewegung in Drahtlossensornetzwerken wird dabei hingewiesen.

Eine systematische Beschreibung verschiedenartiger bewegungskoopertativ zu lösender Problemstellungen ist damit zur optimierungsbasierten, simultanen Trajektorienplanung und Aufgabenzuweisung vorbereitet und mit den Methoden der gemischt-ganzzahligen Optimierung realisierbar.

5 Anwendungen und Ergebnisse

Kapitelinhalt:

Anhand von Benchmarkproblemen kooperativer Mobilität (Abschnitt 5.1 und 5.3) wird für schwer entscheidbare Fragestellungen die Leistungsfähigkeit der vorgestellten diskret-kontinuierlichen Modellierungs-, Approximations- und Optimierungskonzepte systematisch ausgelotet.

Für neuartige, umfassende Fragestellungen realitätsnaher Szenarien in Abschnitt 5.2 werden mit Hilfe linearer gemischt-ganzzahliger Optimierung Antworten vorgestellt.

Die Ergebnisse zeigen sowohl Möglichkeiten als auch Grenzen, der vorgestellten Konzepte zur Analyse, Missionsplanung und MPC-Regelung in kooperativen Mehrfahrzeugsystemen. Abschließend werden Vorschläge zur begleitenden Verwendung von Simulationen präsentiert, die sowohl den Modellierungsprozess der Methoden als auch eine Übertragung von MPC-Reglern auf reale Systeme wesentlich unterstützen.

Zur Demonstration der vorgestellten Modellierungs-, Approximations- und Optimierungskonzepte werden verschiedenartige, aussagekräftige Beispielszenarien untersucht, die in Tabelle 5.1 zusammengefasst sind. Grundlage der Betrachtungen bildet die in Abb. 5.1 dargestellte Auswahl verschiedenartiger Fahrzeuge. Zugehörige charakterisierende Daten sind in Tabelle 5.2 aufgelistet.

Sofern nicht anders angegeben, wurden alle Berechnungen auf einem PC mit Intel® Core™2 CPU (T7200 @ 2.00GHz) und 2 GB RAM durchgeführt. Lineare Optimierungsprobleme wurden dabei in MATLAB[Mat] aufgestellt und über die Schnittstelle CPLEXINT[TB] an den Solver CPLEX[ILO07] übergeben. Zum Vergleich wurden einzelne Probleme mit dem kürzlich vorgestellten Solver GUROBI[GO09] gelöst.

Die Lösung nichtlinearer, gemischt-ganzzahliger Optimierungsprobleme wurde mit Hilfe von BONMIN [WB06] über die Schnittstelle AMPL[AMP] durchgeführt. Referenzlösungen diskret-kontinuierlicher, nichtlinearer Optimalsteuerungsprobleme wurden mit Hilfe der Implementierung MINOCS [Glo05] berechnet, die auf Basis von B&B und direkter Kollokation arbeitet.

Szenario	Systemcharakteristik	Methodischer Fokus
Zielzuweisung an UAVs (S. 75)	<ul style="list-style-type: none"> - versch. Fahrzeugmodelle - Kernproblem kooperativer Erkundung 	<ul style="list-style-type: none"> - Dynamik-Approximationen - Vergleich Referenzlösung
Mobilität in Drahtlossensor-netzwerken (S. 84)	<ul style="list-style-type: none"> - Nachbarschaftsbeziehungen - Konnektivität - koordinierte Bewegung 	<ul style="list-style-type: none"> - Zerlegung in Zielpunkt-generierung und Bahnplanung - Dynamik-Approximation
Kommunikations-beschränkungen (S. 87)	<ul style="list-style-type: none"> - heterogene Fahrzeugflotte - strukturiertes Gebiet - Nachbarschaftsbeziehungen - Konnektivität - koordinierte Bewegung 	<ul style="list-style-type: none"> - Zusammenwirken verschiedener Teilmodelle - PWA-Beschreibung - geometr. Strukturabfolgen
Roboterfußball (S. 93)	<ul style="list-style-type: none"> - geschaltete Dynamik - dynamische Umgebung 	<ul style="list-style-type: none"> - Automatenmodellierung - Echtzeitfähigkeit - Erreichbarkeitsanalyse - Zielfunktionformulierung
Zielbeobachtung (S. 99)	<ul style="list-style-type: none"> - dynamisches Umfeld 	<ul style="list-style-type: none"> - MLD-Beschreibung - Echtzeit MPC-Regelung - Stabilitätsbetrachtung

Tabelle 5.1: Repräsentative Szenarien mit Systemcharakteristik und Methodik.



(a) Quadrocopter „Hector AV“



(b) Luftfahrzeug „MAVinci“



(c) Wasserfahrzeug



(d) Geländefahrzeug „Hector GV“



(e) Fußballroboter

Abbildung 5.1: Im Rahmen der Arbeit betrachtete Fahrzeugmodelle.

Fahrzeug	Daten
Quadrocopter „Hector AV“ (Entwickelt am Fachgebiet Flugsysteme und Regelungstechnik der TU Darmstadt.)	<ul style="list-style-type: none"> - Größe der Grundstruktur: $0.25 \text{ m} \times 0.15 \text{ m}$ - Gewicht: ca. 1.0 kg - max. Geschwindigkeit: $13 \frac{\text{m}}{\text{s}}$ - max. Vorwärtsbeschleunigung: $a_{\max} = 3.13 \frac{\text{m}}{\text{s}^2}$ - max. erlaubte Flughöhe: 150 m - min. Kurvenradius r: $\frac{v^2}{r} \leq a_{\max}$
Luftfahrzeug „MAVinci“ (Micro Air Vehicle) (MAVinci UG, Darmstadt)	<ul style="list-style-type: none"> - Spannweite \times Länge: $1.85 \text{ m} \times 1.3 \text{ m}$ - max. Abfluggewicht: $4,9 \text{ kg}$ - Geschwindigkeit über Grund: $12.5 \frac{\text{m}}{\text{s}}$ bis $25 \frac{\text{m}}{\text{s}}$ - minimaler Kurvenradius: ca. 15 m - max. Flughöhe: 500 m
Geländefahrzeug „Hector GV“ (Entwickelt im Team „Hector“ an der TU Darmstadt)	<ul style="list-style-type: none"> - Größe ($L \times B \times H$): $0.5 \text{ m} \times 0.4 \text{ m} \times 0.48 \text{ m}$ - max. Geschwindigkeit: $1 \frac{\text{m}}{\text{s}}$ (alternativ $5 \frac{\text{m}}{\text{s}}$) - max. Beschleunigung: $2 \frac{\text{m}}{\text{s}^2}$ (alternativ $10 \frac{\text{m}}{\text{s}^2}$) - min. Kurvenradius: 28 cm
Wasserfahrzeug (Entwickelt am Fachgebiet Simulation, Systemoptimierung und Robotik der TU Darmstadt [Eck08].)	<ul style="list-style-type: none"> - Größe ($L \times B \times H$): $1.28 \text{ m} \times 0.27 \text{ m} \times 0.22 \text{ m}$ - Gewicht: 1 kg - Maximalgeschwindigkeit: $0.5 \frac{\text{m}}{\text{s}}$ - Maximale Drehrate: $\frac{1}{3} \text{ s}^{-1}$ - Minimaler Wendekreis: 1.3 m - Maximale Vorwärtsbeschleunigung: $\approx 0.2 \frac{\text{m}}{\text{s}^2}$
Fußballroboter (RoboCup-Team der Universität Stuttgart)	<ul style="list-style-type: none"> - Größe ($L \times B \times H$): $0.5 \text{ m} \times 0.5 \text{ m} \times 0.8 \text{ m}$ - Maximalgeschwindigkeit: $5 \frac{\text{m}}{\text{s}}$ - omnidirektionaler Antrieb

Tabelle 5.2: Daten der betrachteten Fahrzeuge.

5.1 Benchmarkproblem der Zielzuweisung zur kooperativen Erkundung

In diesem Abschnitt werden Grundelemente der vorgestellten diskret-kontinuierlichen Modellierungs- und Approximationskonzepte anhand eines Benchmarkproblems der kooperativen Mobilität evaluiert.

Grundproblem: Zu Beginn und Ende der kooperativen Aufgabe befindet sich eine Gruppe von Fahrzeugen an einer Basisstation. Die Aufgabe besteht darin, eine Menge an Wegpunkten in der Ebene entlang glatter Fahrzeugtrajektorien zu passieren (Abb. 5.2a).

In dieser allgemeinen Form verlangt eine optimale Lösung nach der Berücksichtigung physikalischer Fahrzeugdynamik und erlaubt ohne Optimalitätsverlust keine Abkopplung des diskreten Problemanteils, das heißt der Zuweisung und Sequenzierung der Wegpunkte an die Fahrzeuge. Im Gegensatz zum klassischen Handlungsreisendenproblem hängen die Kosten eines Wegabschnitts stets von den Reihenfolgen *aller* Vorgänger- und Nachfolgepunkte entlang einer Tour ab.

Das Problem besitzt einen kombinatorischen Grundcharakter und ist \mathcal{NP} -schwer. Bereits bei zwei Fahrzeugen und einer Auswahl von $n_c = 5$ Wegpunkten ergibt sich eine

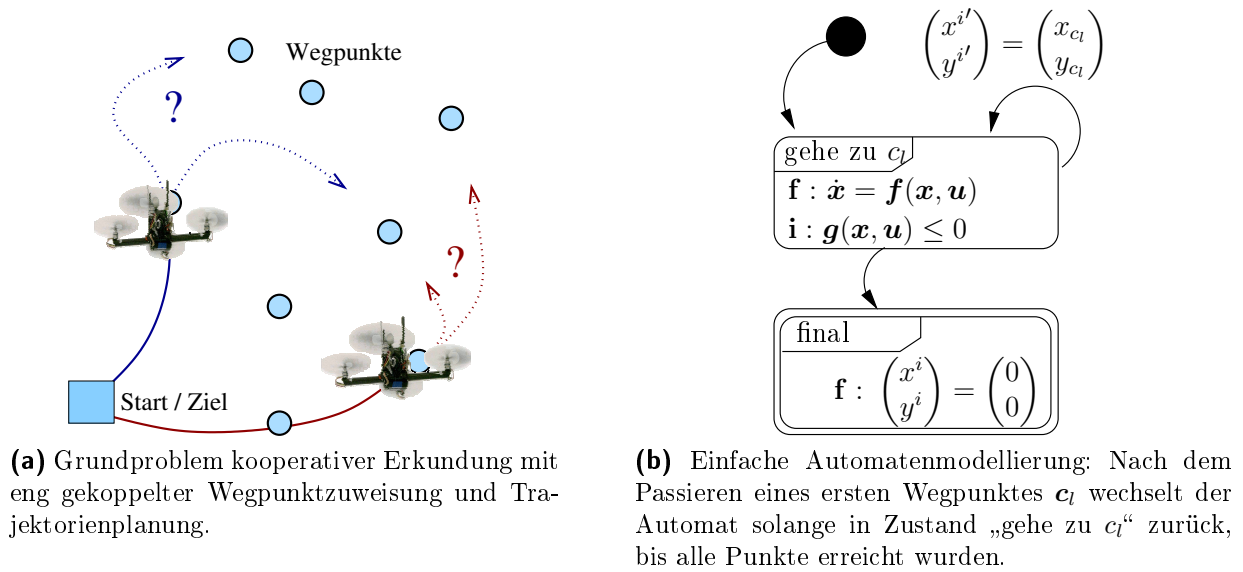


Abbildung 5.2: Grundproblem der kooperativen Erkundung und Automatenmodell.

Zahl von $(n_c + 1)! = 720$ zulässigen Rundreisen durch alle Punkte; diejenigen eingeschlossen, bei denen einem Fahrzeug alle Orte zugeordnet werden. In der Literatur wird das Problem auch unter der Bezeichnung *Problem mehrerer motorisierter Handlungsreisender* untersucht [vSG01, SBF05, GRvS06, SBF09].

5.1.1 Untersuchung des Grundproblems

Zunächst wird folgendes Grundproblem untersucht, um im Abschnitt 5.1.2 dazu spezielle Aspekte der Modellierung und Approximation vergleichend zu bewerten.

Problembeschreibung: Betrachtet wird das Problem sowohl mit $n_v = 2$ als auch mit $n_v = 3$ Fahrzeugen i , beschrieben als Punktmassenmodell 2. Ordnung für $t \in [0, t_f]$:

$$\dot{x}^i(t) = v_x^i(t) \quad , \quad \dot{y}^i(t) = v_y^i(t) \quad , \quad (5.1a)$$

$$\dot{v}_x^i(t) = u_x^i(t) \quad , \quad \dot{v}_y^i(t) = u_y^i(t) \quad . \quad (5.1b)$$

Da kein Schalten in der Bewegungsdynamik stattfindet, kann ein Automatenmodell in sehr einfacher Struktur aufgestellt werden (Abb. 5.2b).

Jeder Wegpunkt c_l muss während der Mission von mindestens einem Fahrzeug passiert werden. Die Mission startet und endet jeweils im Ursprung mit Geschwindigkeit 0:

$$\begin{aligned} x^i(0) &= 0 = x^i(t_f), & y^i(0) &= 0 = y^i(t_f), \\ v_x^i(0) &= 0 = v_x^i(t_f), & v_y^i(0) &= 0 = v_y^i(t_f). \end{aligned}$$

Untersuchte Fragestellungen:

1. Für eine wachsende Zahl an zufällig generierten Wegpunkten $\{c_1, \dots, c_{n_l}\}$ wird durch Analyse der Rechenzeiten eine Einschätzung der Problemkomplexität bei exakter Diskretisierung gewonnen.

2. Approximationsfehler und Optimalitätsverlust werden im Vergleich mit Referenzlösungen qualitativ erfasst, die mit direkter Kollokation und B&B berechnet wurden.

Modellierung: Betrachtet wird ein Gebiet von 7000 m^2 , so dass für alle Wegpunktkoordinaten $(x_{c_l}, y_{c_l}) \in [-30, 70] \times [0, 70]$ gilt. Die Wegpunkte sind zufällig bestimmt derart, dass sie mindestens 10 m voneinander und vom Startpunkt entfernt liegen.

Gemäß den Daten des Quadrocopter-Modells (Tabelle 5.2) werden für die Untersuchungen folgende Beschränkungen der Maximalgeschwindigkeit und maximalen Beschleunigungen verwendet:

$$|v_x^i| \leq 13 \frac{\text{m}}{\text{s}}, \quad |v_y^i| \leq 13 \frac{\text{m}}{\text{s}}, \quad |u_x^i| \leq 3.13 \frac{\text{m}}{\text{s}^2}, \quad |u_y^i| \leq 3.13 \frac{\text{m}}{\text{s}^2}.$$

Das Passieren eines Wegpunktes findet zu einem Schaltzeitpunkt t_k statt. Somit lauten die Beschränkungen, welche Rundreisen durch alle Wegpunkte garantieren,

$$\bigvee_{i=1}^{n_v} \bigvee_{l=1}^{n_l} \left[\begin{pmatrix} x^i(t_k) \\ y^i(t_k) \end{pmatrix} - \begin{pmatrix} x_{c_l} \\ y_{c_l} \end{pmatrix} = 0 \right].$$

Unter Verwendung der binären Variablen $b_l^i(t_k)$ lassen sich die Bedingungen wie folgt als lineare Nebenbedingungen eines Optimierungsproblems formulieren,

$$\forall i, \forall k, \forall l : | -x^i(t_k) + x_l | < \delta + (1 - b_l^i(t_k))M_x \quad (5.2a)$$

$$\forall i, \forall k, \forall l : | -y^i(t_k) + y_l | < \delta + (1 - b_l^i(t_k))M_y \quad (5.2b)$$

$$\forall l \in \{1, \dots, n_l\} : \sum_{i=1}^{n_v} \sum_{k=1}^{n_c} b_l^i(t_k) = 1 \quad (5.2c)$$

$$\sum_{l=1}^{n_l} \sum_{i=1}^{n_v} \sum_{k=1}^{n_c} b_l^i(t_k) = 1. \quad (5.2d)$$

Dabei sind M_x, M_y Big-M-Konstanten und $\delta = 1 \text{ m}$ beschreibt die Genauigkeit, mit der Wegpunkte angefahren werden müssen.

Weiterhin lässt sich die Formulierung des Optimierungsproblems durch zusätzliche Bedingungen verbessern, die (a) gleichwertige globale Optima und (b) nicht erreichbare diskrete Zustände beschreiben.

- (a) Da alle Fahrzeuge dem Modell nach gleich sind und im selben Punkt starten und da das umgekehrte Durchlaufen der gesamten Wegpunktsequenz gleichwertig zum Durchlauf der Ausgangssequenz ist, kann

$$\forall k \in \{1, \dots, \lfloor \frac{n_c}{2} \rfloor\} : b_1^1(t_k) = 0 \quad (5.3a)$$

gesetzt werden, ohne die Qualität der Optimallösung zu beeinträchtigen.

- (b) Über die Geschwindigkeitsbeschränkung $\|(v_x^i, v_y^i)^T\| < v_{max}^i$ lassen sich bei festem Zeitgitter leicht Wegpunkte berechnen, welche zum Zeitpunkt t_k keinesfalls erreichbar sind:

$$\forall l \forall i \forall k \text{ mit } \left\{ \begin{array}{l} \left\| \begin{pmatrix} x_l \\ y_l \end{pmatrix} \right\| > t_k \cdot v_{max}^i \quad \vee \\ \left\| \begin{pmatrix} x_l \\ y_l \end{pmatrix} \right\| > (t_f - t_k) \cdot v_{max}^i \end{array} \right\} : b_l^i(t_k) = 0 \quad (5.3b)$$

Es ist zu erwarten, dass vor allem zu Beginn und am Ende der Mission, wenn die Fahrzeuge aus der Ruhelage starten bzw. in diese zurückkehren, ein hoher Steuereingriff stattfinden muss. Für Untersuchungen mit festem Zeitgitter werden die Punkte t_k deshalb so gewählt, dass der erste Abstand $|t_1 - t_0|$ und der letzte $|t_f - t_{n_c}|$ genau halb so lang ist wie zwischen allen anderen, sonst äquidistant verteilten, Zeiten t_k .

Das resultierende Modell setzt sich damit aus $n_v(4(n_c + 1) + 2n_c)$ kontinuierlichen und $n_v n_l(n_c + 1)$ binären Variablen zusammen, für die $4n_v(3 + n_l(n_c + 1)) + n_l(5n_c - 6) + 1$ Nebenbedingungen berücksichtigt werden. Als Zielfunktion wird bei fester Endzeit t_f der Energieeintrag in das System minimiert

$$\min \sum_{i=1}^{n_v} \int_0^{t_f} \| (u_x^i(t), u_y^i(t))^T \|_2^2 dt, \quad (5.4)$$

wobei wir bei MILP-Modellen mit stückweise konstanter Steuerung den Integranden gemäß Gleichung (4.20) (mit $n_d = 8$ und $\varphi_0 = 0$) approximieren und das Integral durch die Riemann-Summe ausdrücken.

Um eine Abwägung von Genauigkeit und Effizienz der resultierenden Optimierungsprobleme vornehmen zu können, wird in der zweiten Frage der Optimalitätsverlust untersucht. Dieser entsteht im Fall der linearen Dynamik durch die Annahme der stückweise konstanten Steuerung auf festem Zeitgitter. Es genügt dazu, das Problem mit einem Fahrzeug und einer festen Zahl an zufälligen Wegpunkten zu untersuchen, wofür wir Vergleichsrechnungen für unterschiedlich feinen Zeitgitter durchführen.

Zur Berechnung der Referenzlösung mit B&B und direkter Kollokation sind die Zeiten t_k frei und werden durch Optimierung bestimmt. Die Steuerungen werden dabei als stetig angenommen.

Ergebnisse: Für die Untersuchung des oben beschriebenen Problems auf festem Zeitgitter bei exakter Diskretisierung der Dynamik (vgl. Seite 56) wurden für Situationen mit $n_l = 3, \dots, 10$ Wegpunkten jeweils 50 Probleme gelöst. Die Wegpunkte wurden dabei zufällig bestimmt und sowohl der Fall mit $n_v = 2$ als auch mit $n_v = 3$ Fahrzeugen betrachtet.

Die Endzeit der Probleme wurde als $t_f = 5 \cdot \frac{l_{TSP}}{v_{max}}$ festgelegt, wobei l_{TSP} die Länge der kürzesten Standard-TSP-Tour durch alle Punkte ist. Die Zahl der Zeitintervalle wurde passend durch $n_c = \lfloor \frac{2}{n_v} \cdot n_l + 4 \rfloor$ gewählt.

Abbildung 5.4 zeigt die Verteilungen auftretender Rechenzeiten und Abbildung 5.5 den Verlauf einer typischen Lösung. Der Löser CPLEX hat sich dabei – ohne Angabe von Startschätzungen – als effizient gezeigt, wenn ohne Beschränkungen an die Zahl von Schnittebenenenddurchläufen im Wurzelknoten gearbeitet wurde.

Zur Referenzlösung mit B&B und direkter Kollokation wurden die Bewegungsdifferentialgleichungen und Beschränkungen an 7 Lobatto-Punkten pro Phase stückweise durch kubische Polynome approximiert und innerhalb einer äußeren B&B-Schleife wiederholt mit dem angebundenen SQP-Verfahren SNOPT [GMS02] ausgewertet.

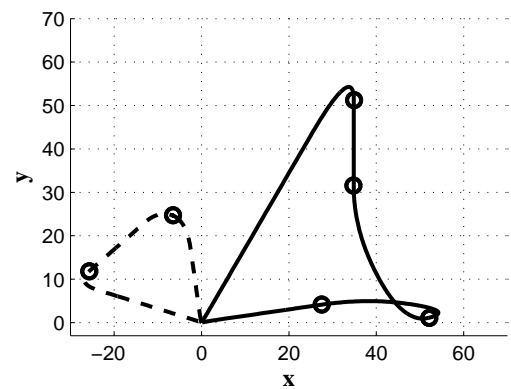


Abbildung 5.3: Optimale Bahnen

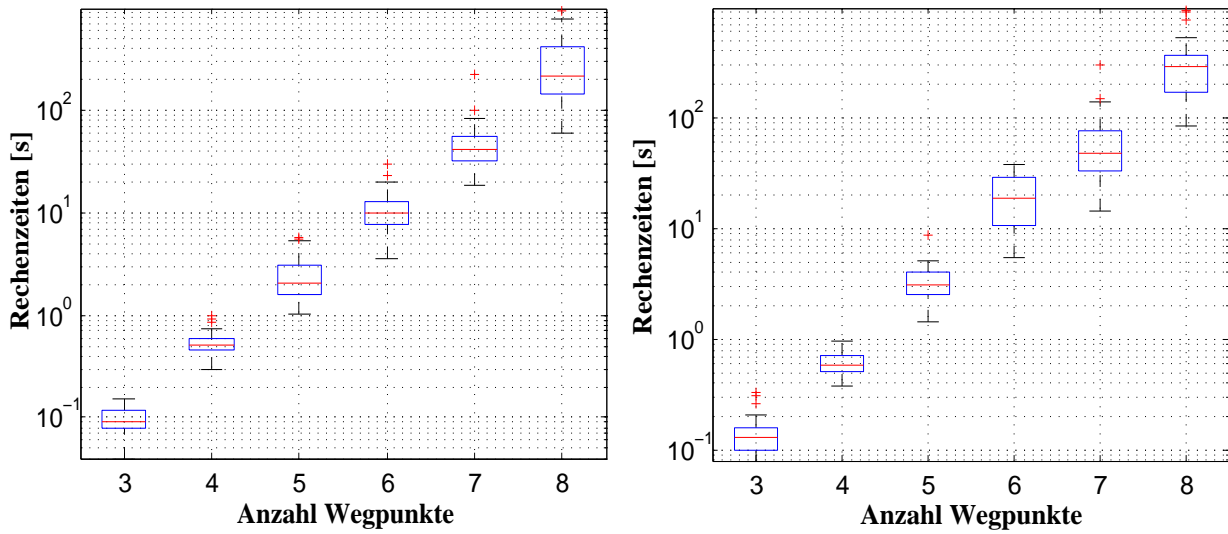


Abbildung 5.4: Rechenzeiten¹ der zeitdiskreten Lösung für 2 (links) und 3 (rechts) Fahrzeuge.

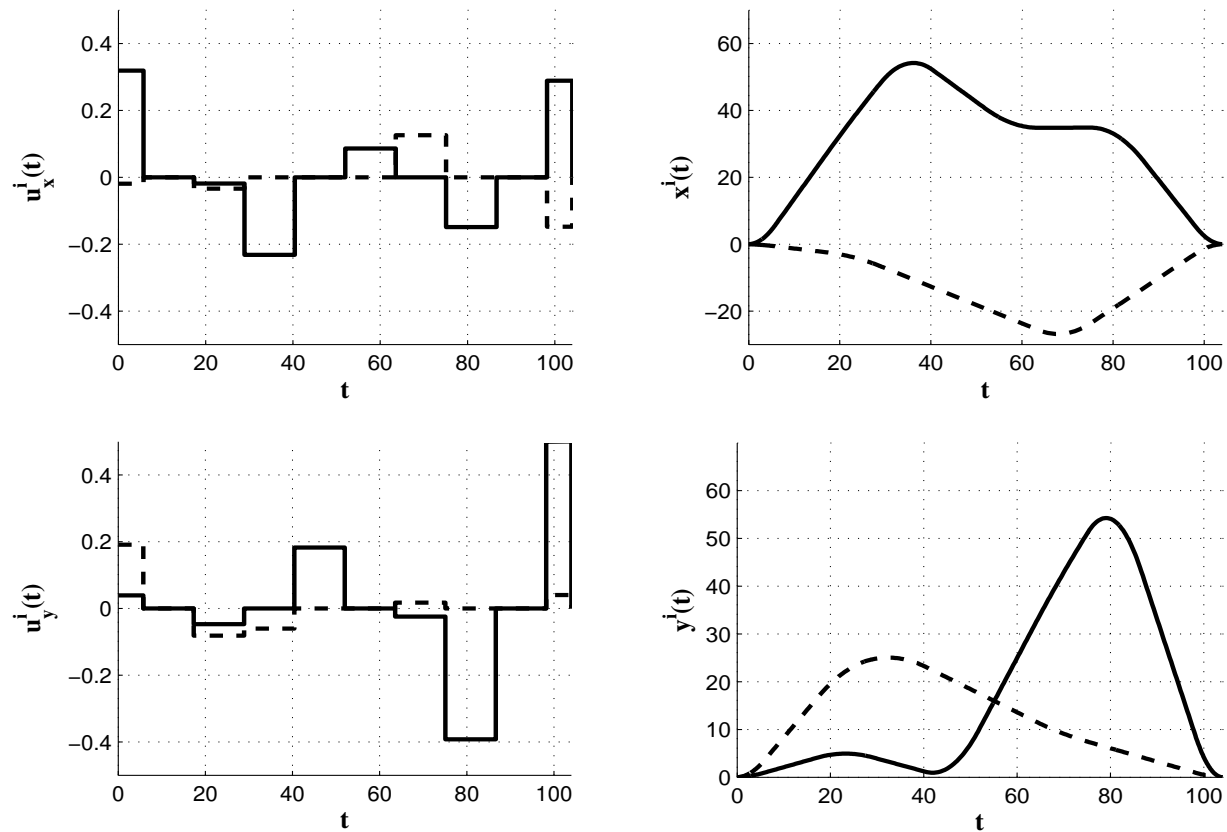


Abbildung 5.5: Steuerungen und Zustandsverläufe der Optimallösung aus Abb. 5.3

¹Die Darstellung in der Kastengraphik stellt als Mittelwert den Median dar und durch einen Kasten den Bereich in dem sich 50 % aller Werte befinden. Die Antennen markieren die Maximalwerte, sofern diese die Grenzen des Kastens um nicht weiter als die 1.5-fache über- bzw. unterschreiten. Weiter entfernte liegende Daten werden als sogenannte Ausreißer (+) markiert.

Für die Optimierung auf dem zeitdiskreten, linearen Systemmodell wurde die Zielfunktion 5.4 in der 2-Norm *nicht* wie in den ersten Beispielen approximiert. Für stückweise konstante Steuerungen beschreibt die Riemann-Summe exakt das Integral der Zielfunktion. Liegt die Zielfunktion somit als Summe über quadratische Ausdrücke vor, ist ein gemischt-ganzzahliges *quadratisches* Minimierungsproblem entstanden, das sich ebenfalls relativ effizient lösen lässt. Die Vergleichbarkeit der Lösung auf Basis einer linearen Modellapproximation mit der Lösung durch die allgemeine, nichtlineare Methode MINOCS auf dem Originalmodell wird damit hergestellt.

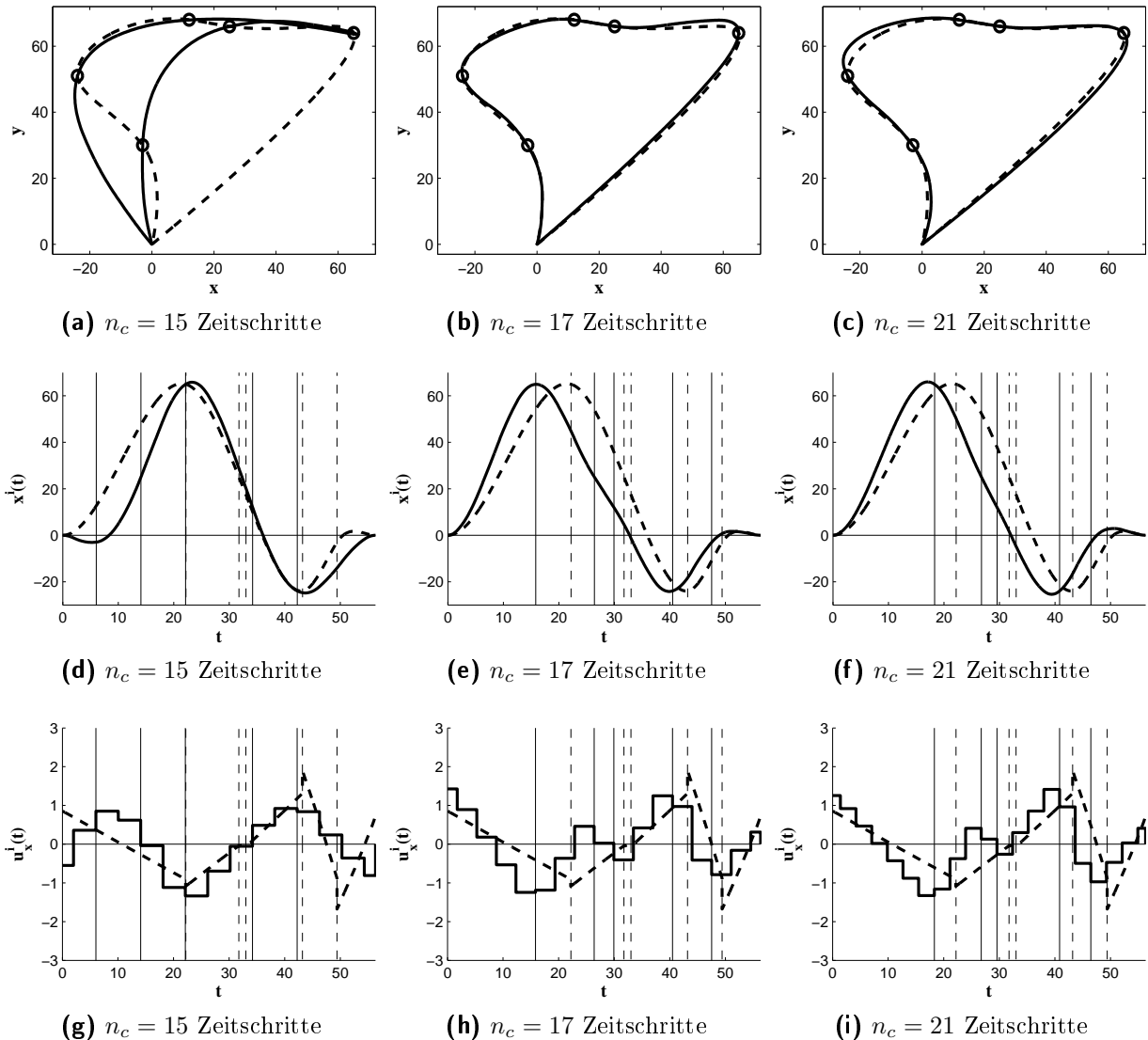


Abbildung 5.6: Mit wachsender Zahl an Diskretisierungsschritten nähern sich sowohl die approximativen Trajektorien (—) der Referenzlösung (---) an. Dies gilt ebenso für die stückweise konstanten Steuerungen und die Schaltzeitpunkte („|“ bzw. „|“).

Für jeweils $n_l = 5$ zufällig gewählte Wegpunkte wird ein nicht äquidistantes Zeitgitter (s. oben) mit variierender Zahl an Zeitschritten bei fester Endzeit betrachtet. Es zeigt sich, dass Approximationslösung und die Referenzlösung sich mit zunehmender Gitterverfeinerung annähern (Abb. 5.7). Abbildung 5.6 zeigt exemplarisch Lösungen eines Beispiels.

Bewertung: Die Rechenzeiten weisen deutlich auf die exponentiell wachsende Komplexität des Problems hin. Im Vergleich zu Berechnungen mit B&B und direkter Kollokation in [Glo05, RGVs09] lässt sich der deutliche Effizienzgewinn beim Übergang auf approximierende Modelle erkennen. Für eine Reihe zufällig generierter Probleme wurde bestätigt (Abb. 5.7), dass sich Lösungen auf dem Ersatzmodell der nichtlinearen Referenzlösung annähern und das Gesamtkonzept somit grundsätzlich zur Generierung von Startschätzungen der kombinierten Methode aus B&B und direkter Kollokation geeignet ist (siehe dazu auch Abschnitt 3.4). Dabei erlaubt die Wahl des Zeitgitters eine einfache Abwägung von Genauigkeit und Berechnungseffizienz. Eine mit n_c exponentiell wachsende Rechenzeit steht dazu der notwendigen Zahl an Mindestschritten gegenüber, die eine zulässige Lösung erst möglich macht.

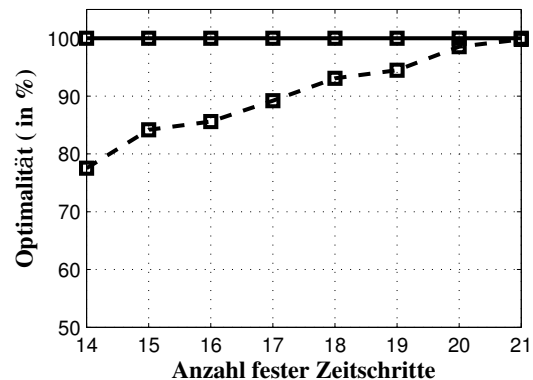


Abbildung 5.7: Für zufällig generierte Wegpunkte zeigt sich, dass das Approximationsmodell und die Referenzlösung der nichtlinearen Methode sich mit feiner werdendem Zeitgitter annähern.

5.1.2 Untersuchung von Detailspekten der Modellierung und Approximation

Problembeschreibung: Für die zeitdiskrete Problemformulierung im vorigen Abschnitt werden der Einfluss weiterer Aspekte des Approximationskonzeptes und Modellerweiterungen betrachtet. Der modulare Aufbau des Gesamtmodells und die Skalierbarkeit der Teilmodelle erlauben dabei eine einfache Anpassung an neue Genauigkeits- und Effizienzanforderungen sowie das Formulieren neuer Fragestellungen.

Untersuchte Fragestellungen: Anhand von Rechenzeiten zufällig generierter Szenarien werden Teilmodelle aus Kapitel 4 untersucht.

- Inwieweit eignen sich Dynamikmodelle aus Abschnitt 4.2.2 in heterogener Zusammensetzung zur effizienten Optimierung?
- Welchen Einfluss hat die Hinzunahme von Kollisions- und Hindernisbeschränkungen (siehe S. 62) auf die Effizienz der Berechnung?

Modellierung: Betrachtet werden Szenarien mit jeweils zwei verschiedenartigen Fahrzeugen, die sich sowohl in ihren Maximalgeschwindigkeiten als auch in der Beschreibung grundsätzlich unterscheiden.

Ein „MAVinci“ Luftfahrzeug wird mit Hilfe eines Punktmassenmodells zweiter Ordnung (bei exakter Zeitdiskretisierung) beschrieben, das um eine Mindestgeschwindigkeit und die Beschreibung zulässiger Kurvenradien gemäß dem Modell von Seite 54 erweitert wird.

Den Geschwindigkeiten des Flugzeugs angepasst wird das Operationsgebiet als $[-400, 900] \times [-30, 900] \text{ m}^2$ gewählt, wobei der Mindestabstand zufällig generierter Ziele mindestens 100 m beträgt.

Ein Quadrocopter wird wie in Abschnitt 5.1.1 als Punktmassenmodell zweiter Ordnung mit exakter Zeitdiskretisierung gewählt.

Das Geländefahrzeug „Hector GV“ wird in der schnellen Variante (max. $5 \frac{m}{s}$) als Einradmodell in der „Dubin-car“ Formulierung aufgenommen. Dabei wird eine Linearisierung um die Punkte $(0, \frac{l}{6}\pi) \in [v_{min}, v_{max}] \times [0, 2\pi]$ mit $l = 0, \dots, 11$ angewendet. Einer besonderen Berücksichtigung bedarf dabei die Periodizität bezüglich der Fahrzeugorientierung. Dazu wird ein Übergang der Orientierung $\theta(t)$ von 2π nach 0 als Ereignis mit Hilfe einer binären Variable formuliert, deren Schalten wiederum einen entsprechend Sprung in $\dot{\theta}(t)$ zulässt.

Die Beschränkungen der Modellverbesserungen in Gleichung (5.3) des Referenzbeispiels können hier nicht ohne weiteres angewendet werden. Die Beschränkung (5.3a) kann lediglich für eines der Fahrzeuge eingebracht werden und Gl. (5.3b) kann nur in einer fahrzeug-spezifischen Formulierung verwendet werden.

Untersucht werden folgende Fälle:

- Die Kombination aus zwei Punktmassenmodellen (2. Ordnung) mit exakter Zeitdiskretisierung. Es werden jedoch unterschiedliche Schranken an die Maximalgeschwindigkeit gelegt: Eines entspricht dem Quadrocopter-Modell („Hector AV“) mit einer Maximalgeschwindigkeit von $13 \frac{m}{s}$ und das andere einem Luftfahrzeug „MAVinci“ mit der Maximalgeschwindigkeit $25 \frac{m}{s}$.
- Die Kombination eines Quadrocoptermodells (als Punktmasse zweiter Ordnung mit exakter Zeitdiskretisierung) und des oben beschriebenen „MAVinci“-Modells.
- Die Kombination des oben beschriebenen Modells eines Geländefahrzeuges „Hector GV“ mit Quadrocoptermodell „Hector AV“.

Für 25 zufällig generierte Sätze von Wegpunkten werden jeweils Wegpunktzuweisung und

optimale Trajektorien für die beschriebenen Fahrzeugkombinationen auf Basis der Approximationsmodelle berechnet. Als Zielfunktion wird die Riemann-Summe von $\sum_{i=1}^{n_v} \int_0^{t_f} \| (u_x^i(t), u_y^i(t))^T \| dt$ gewählt. Die Endzeit wird als $t_f = 3 \cdot \frac{l_{TSP}}{30 \frac{m}{s}}$ festgelegt, wobei l_{TSP} die Länge der kürzesten TSP-Tour durch alle Punkte ist. Die Zahl $n_c = \lfloor \frac{2}{n_v} \cdot n_l + 4 \rfloor$ und die nicht-äquidistante Verteilung der Zeitintervalle wird analog zu Abschnitt 5.1.1 gewählt. Untersuchungen von Kollisions- und Hindernismodellierung (Frage (b)) geschehen auf Basis der zeitdiskreten Modellierung des Abschnitts 5.1.1 (mit Punktmassenmodellen 2. Ordnung und zufällig gewählten Wegpunkten im Bereich $[-30 m, 70 m] \times [0 m, 70 m]$). Das Modell wird um eine Hindernis- und Kollisionsvermeidung gemäß S. 62 erweitert, welche auch die beschriebene Berücksichtigung von Tunnel-Effekten in Folge grober Diskretisierung enthält. Es wird dazu ein quadratisches Hindernis von $15 m \times 15 m$ betrachtet, dessen Position ebenfalls zufällig gewählt wird (mit Abstand von $\geq 10 m$ zu allen Wegpunkten).

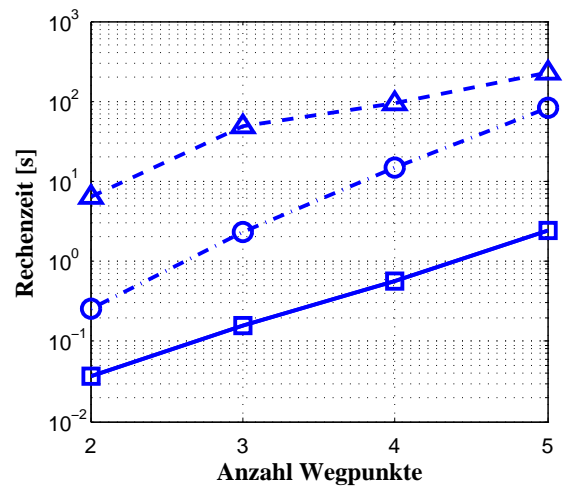


Abbildung 5.8: Rechenzeiten für zwei verschiedene Quadrocopter-Modelle (\square), ein Quadrocopter mit einem „MAVinci“-Luftfahrzeug (\circ) und ein Quadrocopter mit einem Geländefahrzeug „Hector GV“ (\triangle).

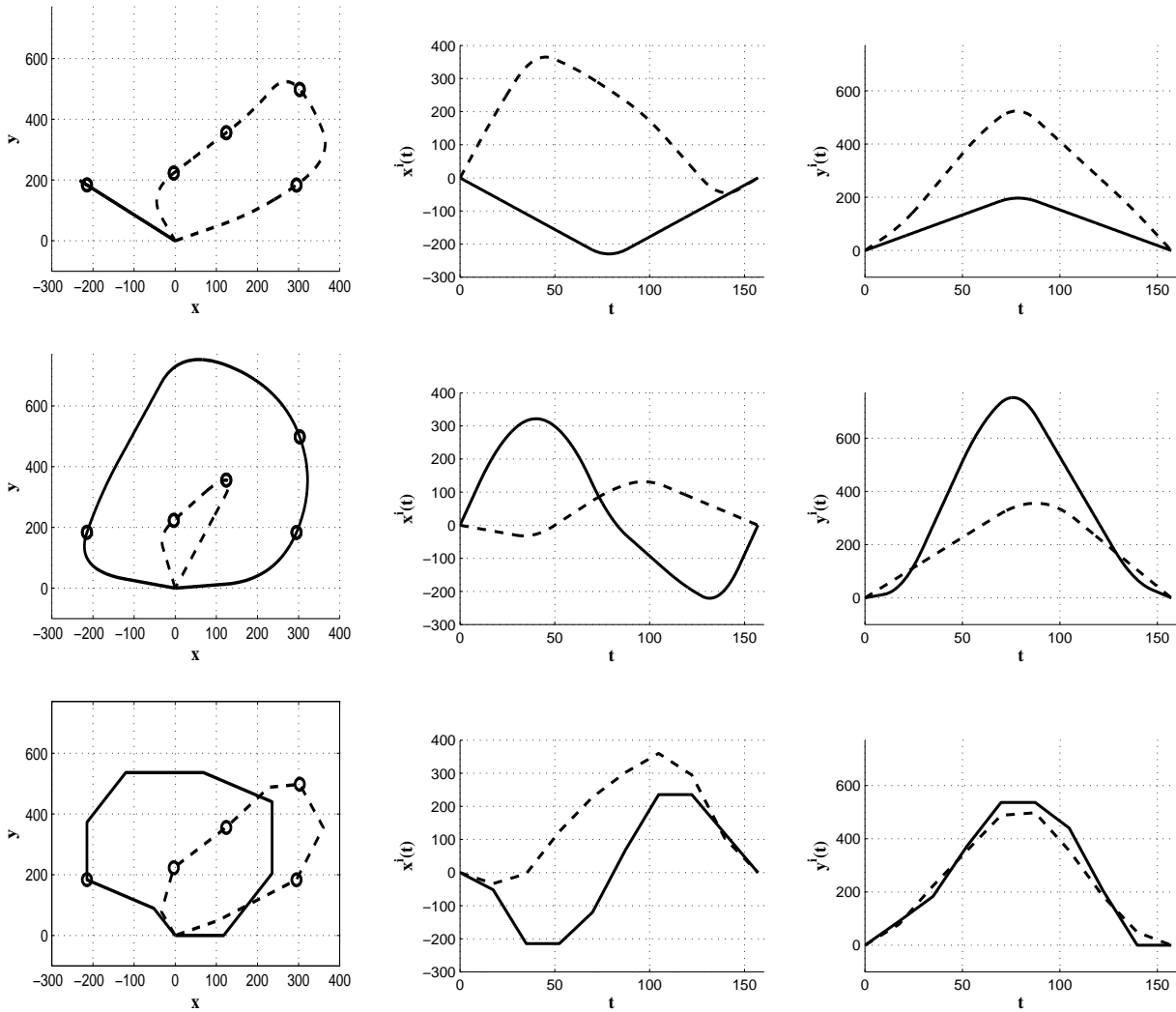


Abbildung 5.9: Untersuchungen eines Szenarios für verschiedene Fahrzeugkombinationen: Zwei Quadcopter mit unterschiedlichen Maximalgeschwindigkeiten (oben), ein Quadcopter mit einem „MAVinci“ (Mitte) und ein Quadcopter mit einem Geländefahrzeug „Hector GV“ in Dubin-car-Modellierung (unten). Es zeigt sich, dass sowohl die Zuweisung der Wegpunkte, als auch die berechneten Bahnen maßgeblich von der Dynamikmodellierung abhängen.

Ergebnisse: Für heterogene Kombinationen von Fahrzeugen ergeben sich die durchschnittlichen Rechenzeiten wie in Abb. 5.8. Exemplarisch zeigt Abb.5.9 resultierende Optimallösungen eines Szenarios für die untersuchten Fahrzeugkombinationen. Dabei sind die spezifischen Merkmale der Fahrzeugdynamiken – etwa die höhere Geschwindigkeit und der große Kurvenradius eines „MAVinci“ – erkennbar und es zeigt sich, dass sowohl die Zuweisung der Wegpunkte, als auch die geplanten Bahnen wesentlich von der Fahrdynamik abhängen.

Zur Untersuchung der Kollisions- und Hindernisbeschränkungen stellten sich auf Basis von jeweils 50 zufällig generierten Szenarien bei Betrachtung von $n_v = 2$ Fahrzeugen die in Abb. 5.10 dargestellten Rechenzeiten ein. Gegenüber den Zeiten in Abb. 5.4 ist eine deutliche Zunahme erkennbar.

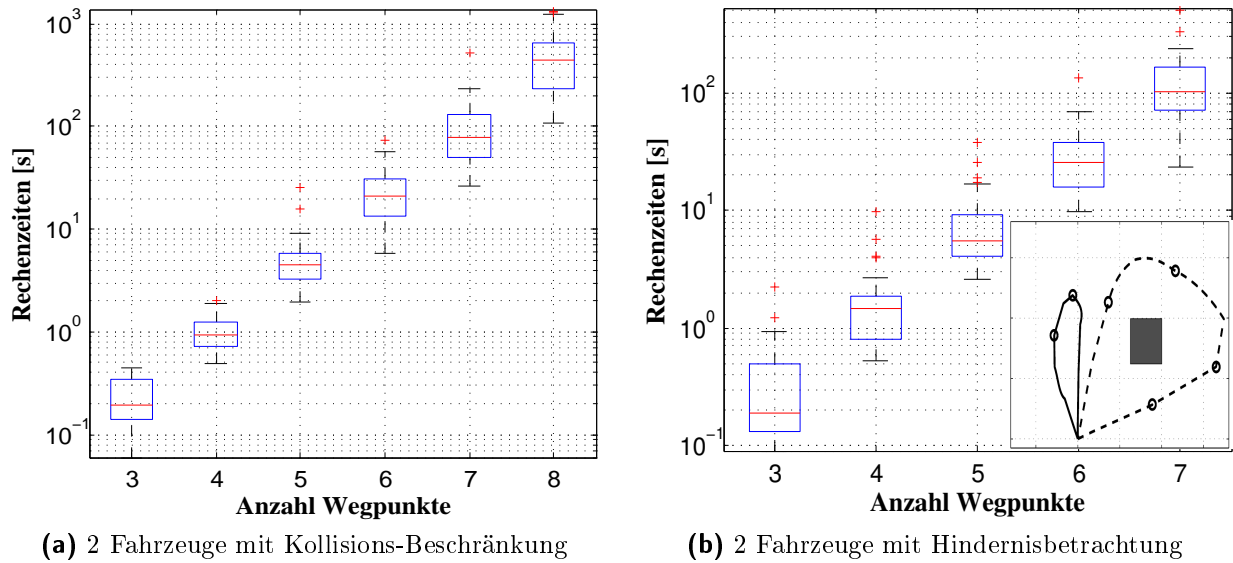


Abbildung 5.10: Berücksichtigung einer Kollisions- und Hindernisvermeidung

Bewertung:

- (a) Neben Punktmassenmodellen erlaubt die Optimierung auf gemischt-ganzzahligen Ersatzmodellen eine Approximationslösung, deren Effizienzvorteil gegenüber nichtlinearen Methoden allerdings mit einer wachsenden Zahl an Betriebspunktlinearisierungen deutlich nachlässt. Aufgrund der zu approximierenden Nichtlinearitäten kann weder die Zahl der Polygone der PWA-Beschreibung noch die Schrittweite im Zeitgitter weiter erhöht werden, ohne Mindestanforderungen an die Genauigkeit zu verletzen.

Die Ergebnisse belegen, dass die Zuteilung und Sequenzierung der Wegpunkte signifikant von der Dynamik abhängt, selbst bei Fahrzeugen mit ähnlichen Maximalgeschwindigkeiten.

- (b) Trotz Hindernissen in einem somit nicht-konvexen Gebiet können global optimale Lösungen effizient berechnet werden.

Gegenüber den Lösungen in Abschnitt 5.1.1 lässt sich allerdings auch ein deutlicher Anstieg der Berechnungskomplexität feststellen. Dies gilt insbesondere für die Betrachtung von Hindernissen unter der Berücksichtigung von Tunnelleffekten, bei denen die Modellerweiterung vorwiegend aufgrund der zusätzlichen diskreten Struktur hier zu einer Verdopplung der durchschnittlichen Rechenzeiten führt.

Allgemein zeigen die Beispiele die Grenzen der Leistungsfähigkeit der Methoden mit wachsender diskreter Problemstruktur. Bedarf ein Dynamikmodell im Zuge der Hybridisierung einer hohen Zahl an Linearisierungen, geht der Effizienzgewinn gegenüber den nichtlinearen Methoden im Allgemeinen verloren.

Weitere Untersuchungen haben ferner gezeigt, dass sich Probleme mit Fahrzeugen unterschiedlicher Maximalgeschwindigkeiten einfacher lösen lassen als Szenarien mit deckungsgleichen Dynamikmodellen und -parametern. Die Ergebnisse in Abschnitt 5.2.2 werden dies bestätigen.

5.2 Mobilität, Kommunikation und Konnektivität

5.2.1 Kooperative Mobilität im Drahtlossensornetzwerk

Problembeschreibung: In einem WSN werden mobile Knoten dazu eingesetzt, Delay-tolerante Daten zu sammeln und an eine Netzwerksenke weiterzuleiten. Das Kernproblem wurde bereits in Abschnitt 4.3.3 beschrieben. Gesucht sind demnach Bahnen und Trajektorien, denen die Fahrzeuge folgen, um an bestimmten Haltepunkten Informationen sowohl mit Sensoren als auch untereinander auszutauschen.

In der Literatur wurden vergleichbare Problemstellungen bislang nur für kleinere Szenarien [TILT09] und häufig nicht allgemein diskutiert. So wird beispielsweise der wichtigen Fall räumlich dicht liegender Sensorknoten nicht gesondert betrachtet. Für größere Probleme wurde in [SG08] ein dem TSPN ähnliches Problem gelöst, aber wiederum ohne Betrachtungen von Überlappungen in den Kommunikationsradien aufeinanderfolgender Kommunikationspunkte.

Die Annahme, dass ein Fahrzeug zum Informationsaustausch anhalten muss, erlaubt eine abgekoppelte Betrachtung der Wegpunktbestimmung, so dass die Lösung des Gesamtproblems nach Algorithmus 3 durchgeführt werden kann. Neben den in Abschnitt 4.3.3 genannten Größen lässt sich der Algorithmus ferner über Zeitkonstanten an die einzelnen Optimierungen anpassen. Der Optimierer liefert dann die beste innerhalb einer Zeitspanne gefundene Lösung an den Benutzer zurück. Für zeitkritische Entscheidungsprozesse lassen sich auf diese Weise effizient sub-optimale Lösungen generieren.

Algorithmus 3: Pfadplanung kooperativer Mobilität im WSN

- 1: /***** Planung der kooperativen, mobilen Datensammlung *****/
 - 2: Bestimme eine reduzierte Menge an Stopppunkten \mathbb{P}_c ;
 - 3: Reduziere Überlappungen der Kommunikationsbereiche;
 - 4: Prüfe, ob eine erneute Reduzierung der Menge \mathbb{P}_c möglich ist;
 - 5: Plane Fahrzeugtrajektorien auf Basis von \mathbb{P}_c ;
-

Untersuchte Fragestellungen: Folgendes neuartige Problem wird untersucht: *Was sind optimale Wegtrajektorien kooperierender Fahrzeuge, so dass die Kommunikationsbereiche aller Sensoren erreicht werden und dass eine Kommunikation über die Fahrzeuge zur Netzwerksenke an ausgezeichneten Zeitpunkten stattfinden kann?*

Die Berechnung von Fahrzeugpfaden kann sehr effektiv als TSP oder – im Fall mehrerer Fahrzeuge – als VRP mit Kapazitäten gelöst werden. Diese Ansätze berücksichtigen aber keinerlei zeitlichen Verlauf der Fahrzeugbewegung. Die Fragestellung hier geht darüber hinaus und verlangt nach einem koordinierten Bewegungsablauf, der es ermöglicht zu erzwingen, dass bestimmte Formationen zu konkreten Zeitpunkten eingenommen werden.

Lösungen zu dieser umfassenden Problemstellung der kooperativen Mobilität und Berücksichtigung verschiedenartiger Kommunikationsbeschränkungen sind dem Autor aus der Literatur nicht bekannt.

Modellierung: Die Berechnung optimaler Haltepunkte erfolgt mit den Modellen aus Abschnitt 4.3.3. Die Annahme, dass Fahrzeuge zur stabilen Kommunikation anhalten müssen,

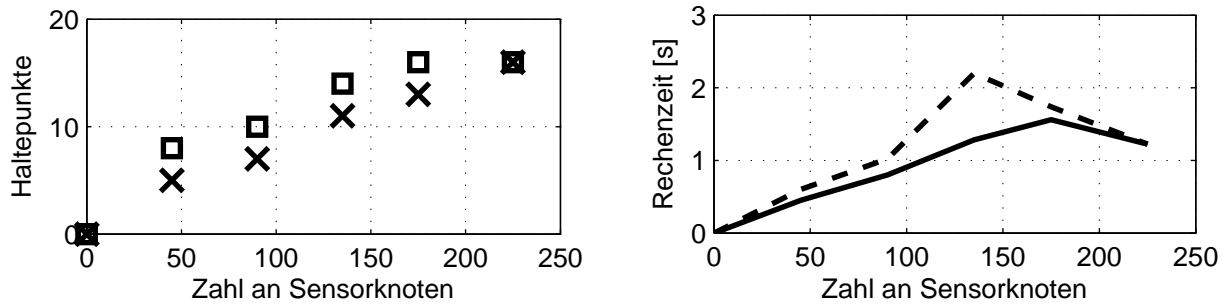


Abbildung 5.11: Ermittelte Zahl an Haltepunkten (links) und Rechenzeiten (rechts). Eine wachsende Zahl zufällig verteilter Sensorknoten wurde auf konstanter Gebietsgröße (—, □) und bei konstanter räumlicher Dichte (—, ×) der Sensorpositionen untersucht.

erlaubt es allerdings, die optimale kooperative Bewegungsstrategie auf Basis eines einfachen Punktmassenmodells erster Ordnung zu berechnen. Die Modellierung der distanzabhängigen, zusammenhängenden Kommunikationsnetzwerkstruktur und Modellverbesserungen werden in Abschnitt 5.2.2 ausführlich dargestellt.

Das konkret untersuchte Szenario besteht aus 225 zufällig in einem rechteckigen Gebiet von $80\text{ m} \times 80\text{ m}$ verteilten Sensorpositionen, die jeweils einen kreisförmigen Bereich stabiler Kommunikation von 10 m besitzen. Es werden drei gleiche Bodenfahrzeuge mit einer Maximalgeschwindigkeit von $2\frac{\text{m}}{\text{s}}$ betrachtet. Die Kommunikationsreichweite der Fahrzeuge untereinander und zur Netzwerksenke betrage 55 m . Die Gebiete paarweiser, verlässlicher Fahrzeugkommunikation werden wie in Abschnitt 4.3.1 mit $n_d = 4$ und $\varphi_0 = 0$ beschrieben.

Über ein festes Zeitintervall von $[0, 200]$ Sekunden wird das approximierte Integral $\int_0^{200} |u_x(t)| + |u_y(t)| dt$ über die Steuerung minimiert.

Ergebnisse: Die Bestimmung der Haltepunkte wurde nach Algorithmus 2 (S. 67) durchgeführt. Dabei wurden die nichtlinearen Optimierungsprobleme mit Hilfe von IPOPT[WB06] über die Schnittstelle AMPL [AMP] berechnet. Für verschiedene zufällige Problemfälle ergaben sich die in Abb. 5.11 gezeigten Zahlen an Haltepunkten.

Für 225 zufällig verteilte Sensorknoten ergaben sich dabei 16 Haltepunkte. Die anschließende simultane Trajektorien- und Reihenfolgenoptimierung wurde für $n_c = 10$ Zeitschritte durch Lösung eines MILP (795 Variablen, 4420 Nebenbedingungen, 10031 nicht-null-Einträge in der Matrix der Nebenbedingungen) berechnet. Die Rechenzeiten lagen mit CPLEX (Einstellung ohne Beschränkungen an die Schnittebenenzahl und mit pseudo-reduzierten Kosten) bei 293s. Der parallel arbeitende Solver GUROBI konnte das Problem in Standardeinstellungen in 45.3s lösen. Der Optimalwert betrug in allen Fällen 93.90. Exemplarisch ist ein Ergebnis in den Abbildungen 5.12 und 5.13 dargestellt.

Bewertung: Zu einer neuen Fragestellung der Unterstützung von Datensammlung und Kommunikation in Drahtlossensornetzwerken mit Hilfe kooperierender Fahrzeuge wurde eine optimierungsbasierte Methode zur simultanen Zuweisung von Wegpunkten und Bahnplanung entwickelt. Die Methode zeigt sich auch für umfangreiche Szenarien – mit dicht liegenden Sensorknoten – als effizient genug, eine optimierungsbasierte Vorabplanung einer Mission durchzuführen.

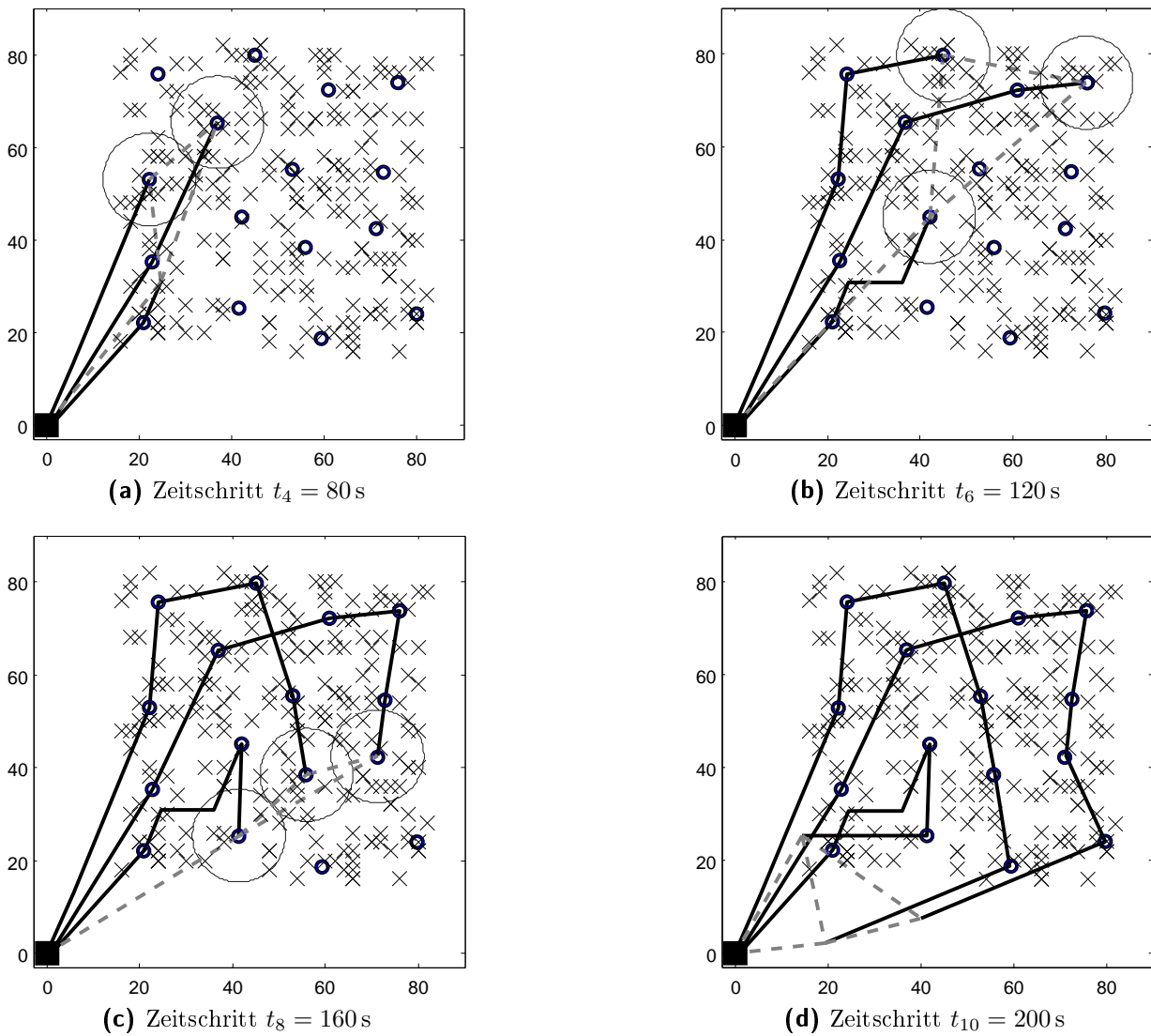


Abbildung 5.12: Zeitlicher Verlauf der kooperierenden mobilen Knoten. Kreise repräsentieren dabei jeweils die Kommunikation mit den Sensoren, während die gestrichelte Linie jeweils eine Kommunikationsverbindung zwischen Fahrzeuge zeigt.

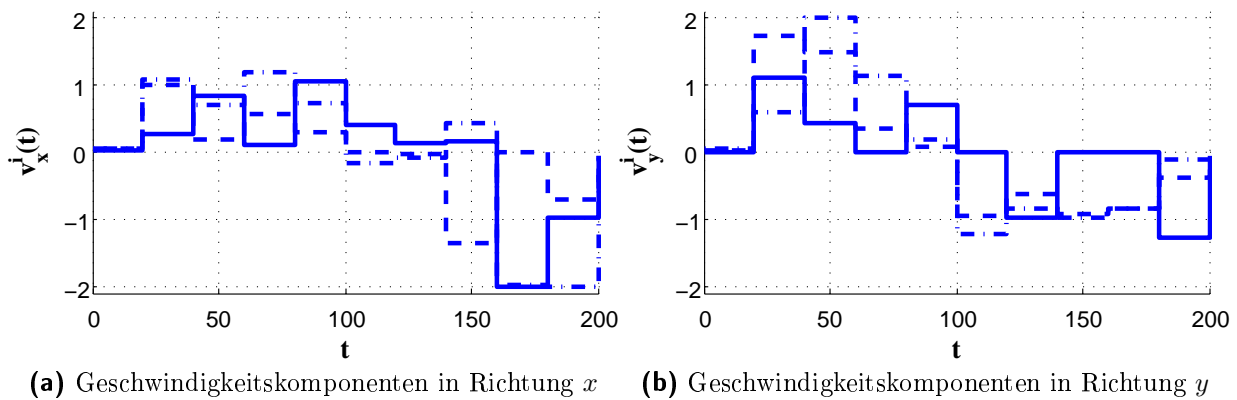


Abbildung 5.13: Errechnetes Geschwindigkeitsprofil zu Abb 5.12.

Kooperationsaspekte mobiler Knoten und Sensoren werden vollständig berücksichtigt. Das berechnete Geschwindigkeitsprofil (Abb. 5.13) gestattet aufgrund der Komponenten $\Delta_{t,k} \cdot (v_x^i(k) + v_x^i(k))$ der Riemann-Summe der Zielfunktion eine nachträgliche Skalierung der Zeitachse ohne Optimalitätsverlust.

Da Kommunikationseigenschaften der Fahrzeuge untereinander berücksichtigt werden, eignet sich die Lösung auch zur Kombination mit dezentralen Regelungsstrategien. Neuplanungen infolge gewonnener Informationen, Ausfällen oder eintretender Störungen können unter den Fahrzeugen während der Mission ausgetauscht werden. Die Fähigkeit zur Kommunikation eröffnet ferner eine Vielzahl weiterer Anwendungen im Gesamtsystem, beispielsweise die Zusammenführung und Verteilung von internen Umgebungsmodellierungen oder spezifischen WSN-Aufgaben.

In Kombination mit einem Datensammlungs-Protokoll konnte in einer detaillierten Simulation die Verbesserung durch die Bahnplanungsstrategie gezeigt werden ([KSA⁺10]).

5.2.2 Kooperative Mehrfahrzeugsysteme mit Kommunikationsbeschränkungen

Nachdem die Anwendung der Methoden auf das Kernproblem kooperativer Erkundungsmissionen in Abschnitt 5.1.1 evaluiert wurde, wird nun exemplarisch eine realitätsnahe Anwendung betrachtet, in der verschiedene wesentliche Aspekte eines kooperativen Mehrfahrzeugszenarios simultan berücksichtigt werden.

Die neuartige Problemstellung berücksichtigt dabei für eine heterogene Fahrzeugflotte sowohl die Gegebenheiten des Operationsbereichs als auch gesicherte Kommunikationsstrukturen der Fahrzeuge untereinander.

Die Lösung des Problems liefert eine optimale Aufteilung von Beobachtungszonen, zusammen mit optimierten Fahrzeugtrajektorien einer synchronisierten Fahrdynamik. Die Lösung kann somit sowohl zur Planung als auch zur Abschätzung der Dauer und des Energieaufwands konkreter Missionen genutzt werden. Die Kombination dieser Merkmale beschreibt in ihrer Gesamtheit die wesentliche Charakteristik kooperativer Erkundung unter realen Bedingungen. Die Fragestellungen in den Abschnitten 5.1.1 und 5.2.1 sind Spezialfälle dieser Betrachtungen.

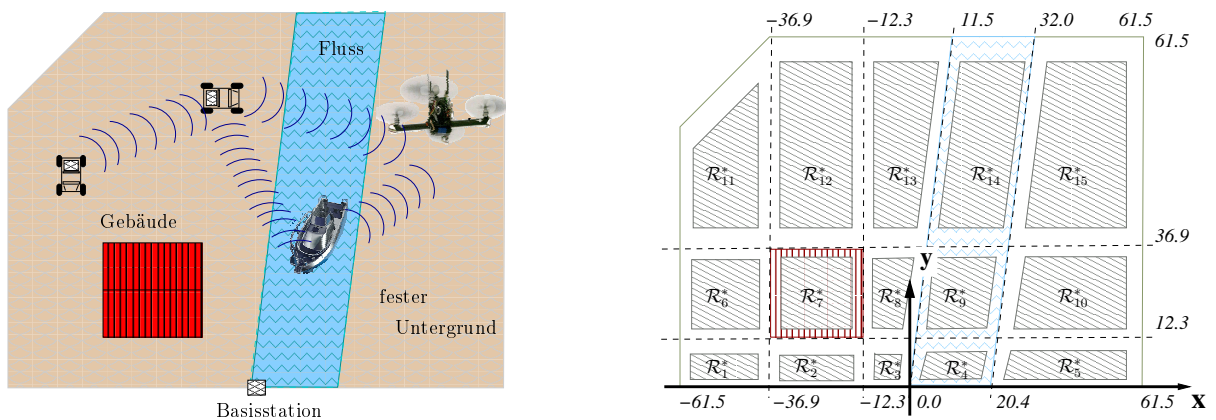


Abbildung 5.14: Einsatzgebiet mit Gebäude und Fluss (links); Abstrahierte Einteilung in Erkundungszonen (rechts).

Problembeschreibung: Betrachtet wird ein abgegrenztes Gebiet im Außenbereich, das eine Strukturiertheit aufweist, die sich über Gebäude, Hindernisse oder Gewässer (vgl. Abb. 5.14) charakterisiert. Zur Erkundung dieses Gebietes stehen verschiedenartige Fahrzeuge bereit; zwei Landfahrzeuge, ein Boot und ein Luftfahrzeug. Alle Fahrzeuge sind in der Lage – beispielsweise über WLAN – Informationen auszutauschen und zu übertragen.

Die Fahrzeuge sind ihrer Bauart nach nur in einem Teil des Gesamtgebietes einsetzbar. Dies wird auf Basis einer Partition durch konvexe Teilmengen \mathcal{R}_l des Aktionsgebietes beschrieben, die den Fahrzeugen zugeordnet werden können (vgl. Tabelle 5.3). Die Erkundungsaufgabe ist definiert durch eine Menge fest vorgegebener Teilbereiche $\mathcal{R}_l^* \subset \mathcal{R}_l$. Von der Fahrzeuggruppe wird verlangt, dass jeder Teilbereich \mathcal{R}_l^* mindestens einmal während der Mission passiert wird. Um die aufgenommenen Daten auszutauschen und um gegebenenfalls während der Aufgabendurchführung eine Anpassung der Planung vornehmen zu können, ist ein beständiges, verlässliches Funk-Kommunikations-Netzwerk erforderlich. Dieses resultiert aus einer paarweisen Nachbarschaftsbeziehung der Fahrzeuge, kombiniert mit einer Forderung an die Netzwerktopologie.

Fahrzeug (Index i)	Einsatzgebiet \mathcal{R}_l
Bodenfahrzeug (1, 2)	1, 2, 3, 5, 6, 8, 10,
„Hector GV“	11, 12, 13, 15
Wasserfahrzeug (3)	4, 9, 14
Quadrocopter (4)	1, 2, ... , 15
„Hector AV“	

Tabelle 5.3: Fahrzeuge und Einsatzgebiete.

Untersuchte Fragestellungen: Es wird die Zusammenführung verschiedenartiger, relevanter Aspekte eines realen Szenarios in einem Modell betrachtet. Es geht darum, die Leistungsfähigkeit der Modellierung approximativer, gemischt-ganzzahliger linearer Modelle weiter zu charakterisieren.

Konkret suchen wir rechnerisch eine Antwort auf die Frage, wie $n_v = 4$ Fahrzeuge eine Auswahl an Teilbereichen \mathcal{R}_l^* eines Gebietes (Abb. 5.14) innerhalb einer festen Zeitspanne $[t_0, t_f]$ optimal besuchen. Dabei werden energieminimale Zustandsverläufe aller Fahrzeuge gesucht, die garantieren, dass die Fahrzeuge während der gesamten Mission eine beständige Funkkonnektivität behalten.

Die Resultate dieses Abschnitts erlauben einerseits Abschätzungen von Eigenschaften einer optimalen Aufgaben- und Trajektorienplanung, andererseits aber gestatten sie es auch allgemein Kriterien abzuleiten, wann die Methodik in zeitkritischen Anwendungen zum Einsatz gebracht werden kann und wo die Grenzen dessen liegen.

Die Herausforderung dieser Fragestellung liegt in den zusammengeführten verschiedenartigen Aspekten kooperativer Mehrfahrzeugsysteme, die zu einem hochdimensionalen Optimierungsproblem mit dominierender diskreter Struktur führt.

Modellierung: Aufgrund der Komplexität des Modells und zahlreicher Nicht-Konvexitäten im Steuer-Zustands-Raum ist eine effiziente Berechnung mit Standardmethoden der nichtlinearen diskret-kontinuierlichen Optimalsteuerung praktisch nicht möglich und wir beschränken uns darauf, eine approximierende, zeitdiskrete, lineare Beschreibung vorzustellen. Da das System keine geschaltete Dynamik aufweist, wird ferner auf eine Modellierung mit hybriden Automaten verzichtet und direkt das zu lösende gemischt-ganzzahlige lineare Optimierungsproblem beschrieben.

Betrachtet wird das System für eine Zeitspanne von 80 s, die in $n_c = 12$ Teilintervall-

le aufgeteilt wird. Da zu Beginn und Ende der kooperativen Mission große Änderungen der Zustandsgrößen zu erwarten sind, wird für das erste und letzte Teilintervall die Länge $\Delta_{t,1} = \Delta_{t,n_c} = \frac{80}{22}$ s und für alle übrigen $\Delta_{t,k} = \frac{80}{11}$ s festgesetzt. Da hier ferner Änderungen der Flughöhe des Luftfahrzeuges keine Rolle spielen, wird das Problem in der Ebene betrachtet.

Zur Beschreibung der Bewegungsdynamik der beteiligten Fahrzeuge werden Punktmaschenmodelle zweiter Ordnung mit exakter Zeitdiskretisierung (vgl. Abschnitt 4.2.2, Parameter siehe Tabelle 5.2) gewählt. In Anbetracht der Verhältnisse von Gebietsgröße zu Maximalgeschwindigkeiten, dürfen damit hinreichend genaue Ergebnisse erwartet werden.

Start- und Endbedingungen sind durch $-3 \leq x^i(1) \leq 3$, $x^i(n_c + 1) = 0$ und $v_x^i(1) = 0 = v_x^i(n_c + 1)$ für alle Fahrzeuge i gegeben; analog für y^i und v_y^i .

In dem untersuchten Beispiel sind Steuerungen \mathbf{u} und Geschwindigkeiten \mathbf{v} mit den Werten aus Tabelle 5.2 in der $\|\cdot\|_2$ -Norm beschränkt. Die Norm wird mit $n_d = 4$ und $\varphi_0 = 0$ gemäß Abschnitt 4.3.1 in das MILP aufgenommen.

Alle Zustandsgrößen \mathbf{x} seien an den Schaltzeitpunkten t_k stetig,

$$\forall i \in \{1, 2, 3, 4\}, \forall k \in \{1, 2, \dots, n_c\} : \begin{pmatrix} x^i(t_k + 0) \\ y^i(t_k + 0) \\ v_x^i(t_k + 0) \\ v_y^i(t_k + 0) \end{pmatrix} = \begin{pmatrix} x^i(t_k - 0) \\ y^i(t_k - 0) \\ v_x^i(t_k - 0) \\ v_y^i(t_k - 0) \end{pmatrix} .$$

Die Beschreibung des Gebietes (Abb. 5.14) geschieht durch die fahrzeugspezifische Abstraktion einer Graphenstruktur, wie dies im Abschnitt 4.3.2 bereits beschrieben wurde. Durch binäre Variablen $b_l^i(k)$ und einfache lineare Ungleichungen lassen sich jedem Fahrzeug die Knoten \mathcal{R}_l (siehe Tabelle 5.3) seiner Graphenstruktur zuordnen. Damit wiederum lässt sich zu jedem Knoten \mathcal{R}_l die Menge anschließender Kanten – resultierend aus der gegebenen Maximalgeschwindigkeit des Fahrzeugs – leicht mit einfachen Ungleichungen in das Modell aufnehmen. Im Zuge der Optimierung erweisen sich diese als sehr effizienzverbessernd.

Im gegebenen Beispiel stellt das Gebäude (Region \mathcal{R}_7) für die Bodenfahrzeuge ein echtes Hindernis dar, zu dessen Beschreibung die verbesserte Hindernismodellierung von S. 62 angewendet wird.

Die Beobachtungszonen \mathcal{R}_l^* seien über die Grenzen der Regionen \mathcal{R}_l und einem Abstand $d_{\mathcal{R}^*} = 4$ m dergestalt definiert, dass

$$\begin{pmatrix} x \\ y \end{pmatrix} \in \mathcal{R}_l^* \Rightarrow \begin{pmatrix} x \pm d_{\mathcal{R}^*} \\ y \pm d_{\mathcal{R}^*} \end{pmatrix} \in \mathcal{R}_l .$$

Mit binären Variablen $b_l^i(k) \in \{0, 1\}$ ($k = 1, \dots, n_c$, $i = 1, \dots, n_v$, $l = 1, \dots, n_R$) lässt sich die Aussage $b_l^i(k) = 1 \Rightarrow (x^i(k), y^i(k))^T \in \mathcal{R}_l^*$ durch

$$\Gamma_l \begin{pmatrix} x^i(k) \\ y^i(k) \end{pmatrix} - \gamma_l + b_l^i(k) \mathbf{M}_l \leq \mathbf{M}_l$$

formulieren. Γ_l und γ_l beschreiben die Polygone \mathcal{R}_l (vgl. Definition 6) und \mathbf{M}_l sind Konstanten des Big-M-Ansatzes derart, dass gilt

$$\mathbf{M}_l \leq \max_{x,y} (\Gamma_l \begin{pmatrix} x^i(k) \\ y^i(k) \end{pmatrix} - \gamma_l) .$$

Die Ausdrücke

$$\left[\forall l : \sum_{k=1}^{n_c} \sum_{i=1}^{n_v} b_l^i(k) = 1 \right] \wedge \left[\sum_{l=1}^{n_R} \sum_{k=1}^{n_c} \sum_{i=1}^{n_v} b_l^i(k) = 1 \right]$$

garantieren, dass jede Beobachtungszone \mathcal{R}_l^* im Laufe der Mission von mindestens einem Fahrzeug besucht wird.

Paarweise Konnektivität wird wie in Abschnitt 4.3.1 mit $n_d = 4$ und $\varphi_0 = 0$ beschrieben. Ein Wechsel der resultierenden Netzwerktopologie ist zu jedem Zeitschritt t_k erlaubt. Mit den Bedingungen

$$\left[\forall i_1 : \sum_{\substack{i_2=1 \\ i_2 \neq i_1}}^{n_v} b_{i_1, i_2}(k) \geq 1 \right] \wedge \left[\sum_{i_1=1}^{n_v} \sum_{\substack{i_2=1 \\ i_2 \neq i_1}}^{n_v} b_{i_1, i_2}(k) \geq 3 \right] \quad (5.5)$$

wird gefordert, dass die Positionen der $n_v = 4$ Fahrzeuge einen zusammenhängenden, distanzabhängigen Graphen (vgl. auch Abschnitt 4.3.2) aufspannen.

Durch weitere Beschränkungen an den zeitlichen Verlauf von Topologiewechseln lässt sich die interne Kommunikationsfähigkeit der Fahrzeuge weiter verbessern. Dieser Aspekt wird an dieser Stelle aber nicht weiter verfolgt.

Kollisionen der Fahrzeuge untereinander können allenfalls für Bodenfahrzeuge auftreten, zeigen aber bei der gegebenen Größe des Einsatzgebietes nur einen vernachlässigbaren Einfluss auf die Aufgabenplanung. Die Kollisionsvermeidung der Fahrzeuge wird deshalb im Modell nicht berücksichtigt und sollte im realen System dezentral erfolgen.

Die in Abschnitt 4.1 angesprochenen Verbesserungen der Modellierung lassen sich wie folgt einbringen:

1. Aufgrund der gemeinsamen Start- und Zielregion aller Fahrzeuge und aufgrund des symmetrischen Zeitgitters liegt eine Symmetrie in den möglichen Reihenfolgen der zu besuchenden Teilgebiete vor. Die zusätzliche Forderung

$$\forall k = 1, \dots, \lfloor \frac{n_c}{2} \rfloor : b_7^i(k) = 0$$

besagt, dass das Flugzeug in den ersten Zeitschritten nicht die Daten im Gebiet \mathcal{R}_7 sammeln wird. Die Symmetrie wird dadurch beseitigt, ohne dabei die Qualität der Lösung zu beschränken.

2. Aufgrund der gegebenen Fahrzeuggeschwindigkeiten ist es keinem der Fahrzeuge möglich, bereits während der ersten Zeitspanne $[t_0, t_1]$ eine der Regionen \mathcal{R}_1 , \mathcal{R}_6 , \mathcal{R}_{11} , \mathcal{R}_{12} , \mathcal{R}_{13} , \mathcal{R}_{14} oder \mathcal{R}_{15} zu erreichen. Mit dieser Information lässt sich der diskrete Suchraum präziser ohne Qualitätsverlust formulieren.
3. Weiterhin bleiben durch zwei identische Bodenfahrzeuge in selber Ausgangsposition gleichwertige Lösungen möglich, von denen eine mit $b_3^1(1) = 0$ eliminiert wird.

Als Zielfunktion wird eine Formulierung gewählt, welche einen Energieeintrag ins System minimiert,

$$\min_{\mathbf{u}} \sum_{k=1}^{n_c} \Delta_{t,k}^2 \left\| \begin{pmatrix} u_x^i(k) \\ u_y^i(k) \end{pmatrix} \right\|_2^2,$$

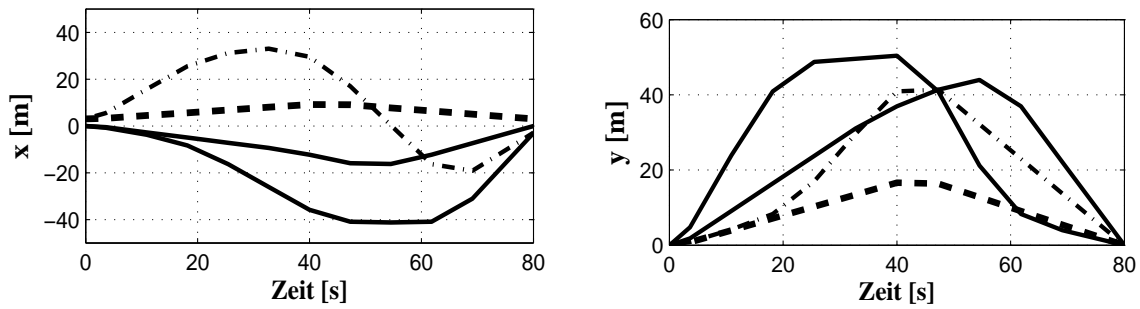


Abbildung 5.15: Positionen x (links) und y (rechts) der Fahrzeuge.

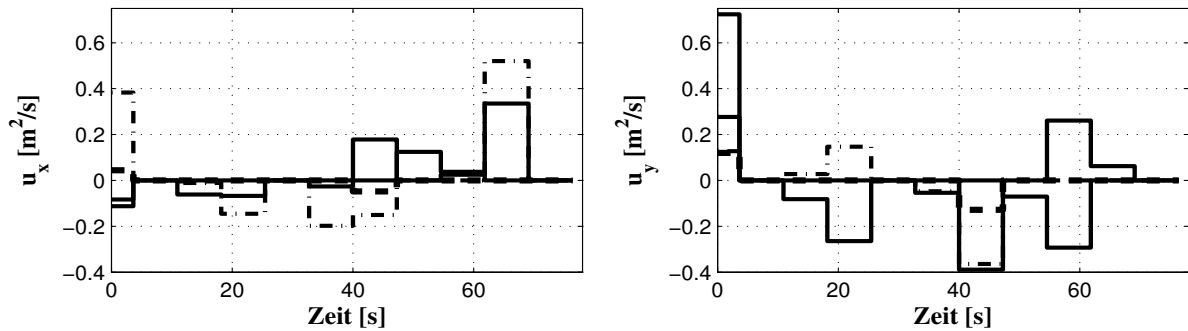


Abbildung 5.16: Beschleunigungen u_x (links) und u_y (rechts) der Fahrzeuge.

wobei die Norm gemäß Gleichung (4.20) mit $n_d = 8$ und $\varphi_0 = 0$ approximiert wird.

Die Modellierung resultiert in einem MILP mit 1558 Variablen (davon 1158 binäre) und mit 10526 Ungleichungsnebenbedingungen. Die dünn besetzte Matrix der Nebenbedingungen hat dabei 25647 von Null verschiedene Einträge.

Ergebnisse: Das MILP wurde – wiederum ohne Angabe von Startschätzungen – mit CPLEX in einer durchschnittlichen Rechenzeit von 470s gelöst. Der sich ergebende Optimalwert lag bei 43.72. Eine weitere Untersuchung mit einem vereinfachten Modell mit Punktmassenmodellen der Ordnung 1 ergab ähnliche Ergebnisse bei einer Rechenzeit von etwa 22s.

Dabei hat sich die Optimierung mit CPLEX als besonders effizient gezeigt, wenn es ohne Beschränkungen an die Zahl von Schnittebenendurchläufen im Wurzelknoten, ohne Vorabuntersuchung zu branching-Regeln und mit pseudo-reduzierten Kosten arbeitet.

Die Abbildungen 5.16 und 5.15 zeigen die optimalen MILP-Steuerungen und Positionsverläufe in x - und y -Richtung. In den Abb. 5.17 sind die resultierenden Pfade in der x - y -Ebene mit den unterlagerten Konnektivitäten graphisch dargestellt.

Bewertung: Das Beispiel zeigt, dass es MILP-Formulierungen erlauben, für eine Vielzahl praxisrelevanter Merkmale eine approximative Optimallösung zu berechnen. Insbesondere sind dies Aspekte heterogener Fahrzeugflotten, eines strukturierten Gebietes sowie Kommunikationsbeschränkungen, die modular und skalierbar in eine einheitliche Modellierung zusammengefasst wurden.

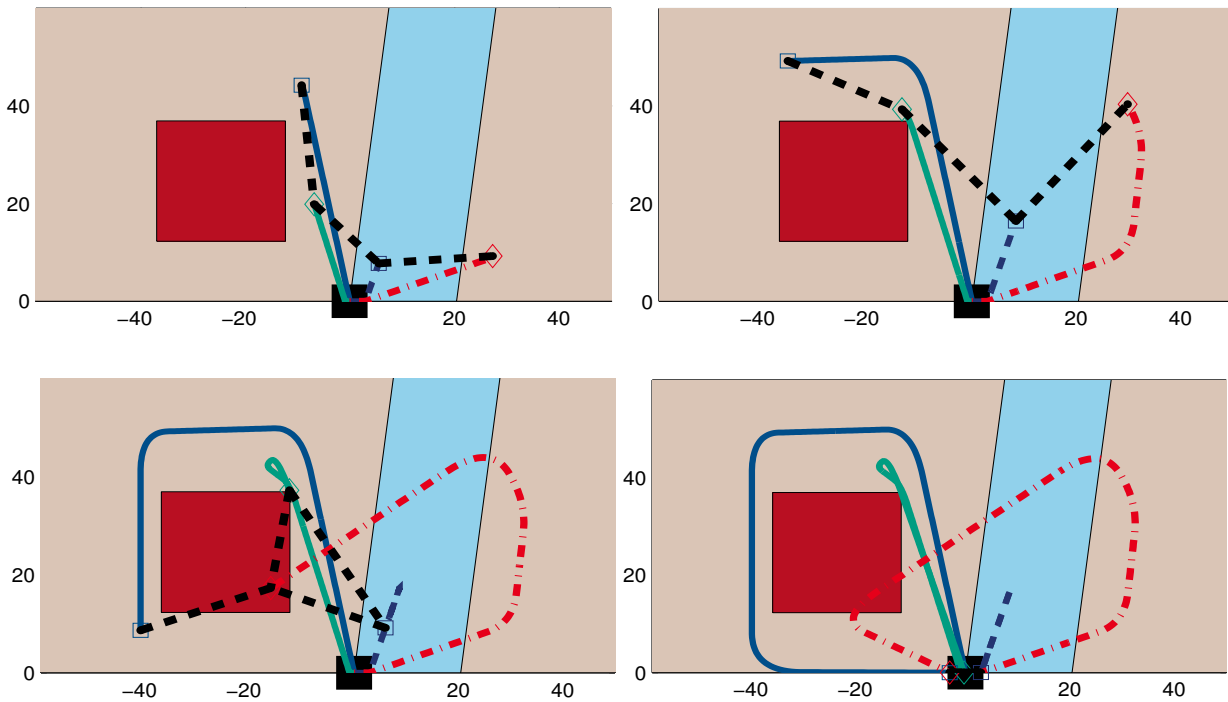


Abbildung 5.17: Situation nach den Zeitpunkten $t_4 = 25.5\text{s}$, $t_7 = 47.3\text{s}$ (oben) und $t_{10} = 54.5\text{s}$, $t_{nc} = 76.4\text{s}$ (unten): Eines der Landfahrzeuge (—, —) fährt um das Gebäude herum, während das Luftfahrzeug (-.-.-) den Fluss überquert und über das Gebäude hinwegfliegt. Alle Fahrzeuge halten eine Kommunikationsverbindung (-.-.-) zum Boot (-.-.-).

Dies macht deutlich, dass die Methodik nicht auf einen engen Kreis von Anwendungen beschränkt ist, sondern zu diversen bewegungs-kooperativ geprägten Problemstellungen hin ausgebaut werden kann. So etwa erlaubt eine Erweiterung der Zielfunktion auch eine Optimierung der Stabilität des Netzwerkes, indem Topologiewechsel berücksichtigt werden.

Die Rechenzeiten zeigen allerdings auch, dass man in Grenzbereiche vordringt und dass eine sorgfältige Abwägung der Modellgenauigkeit und Effizienz getroffen werden muss.

5.3 Kooperatives Verhalten in dynamischer Umgebung

Dieser Abschnitt stellt einerseits Ergebnisse MILP-basierter Methoden zur Analyse des Systemverhaltens vor und andererseits eine effiziente, echtzeitfähige MPC-Regelung für ein repräsentatives, mobilitäts-kooperatives Szenario. Das charakterisierende Merkmal der gewählten Beispiele stellt die dynamische Umgebung mobiler Roboter dar. Jede Aufgabenzuweisung und Trajektorienplanung ist dabei stark situationsabhängig und muss somit zur Laufzeit auf sich ändernde Verhältnisse der Umgebung reagieren können.

Da die betrachteten Systeme hier stets mit ihrer Umwelt in enger Wechselwirkung stehen – insbesondere, wenn eine aktive Manipulation der Umgebung einen Teil der Aufgabe darstellt –, ist eine Rückführung von Sensordaten in die Steuerungsarchitektur essentiell. Nur so können unerwartete Veränderungen der Umgebung, bedingt etwa durch frühere Messungenauigkeiten, durch externe Einflüsse oder durch eine fehlerhafte Aktion eines

Team-Mitgliedes, hinreichend Berücksichtigung finden und passende Reaktionen eingeleitet werden. Die Klärung der daraus resultierenden Anforderungen spiegelt die Grundfrage der MPC-Regelung wider (vgl. Abschnitt 3.3.3), deren Anwendbarkeit wir für die gewählten Beispiele ausloten.

5.3.1 Taktik und Trajektorienplanung im Roboterfußball

Roboterfußball ist ein abgeschlossenes, sicheres Szenario eines mobilen Mehrrobotersystems und ein anschauliches Beispiel kooperativen Roboterhaltens. Demzufolge ist Roboterfußball als wichtiger vorbereitender Schritt zur Untersuchung in offenen, unbekannten und unsicheren Umgebungen allgemein wissenschaftlich akzeptiert. Das Problem besitzt auch aus Sicht der Regelungstheorie eine verallgemeinerbare Charakteristik mit indirekt und direkt steuerbaren sowie nicht steuerbaren mobilen Objekten.

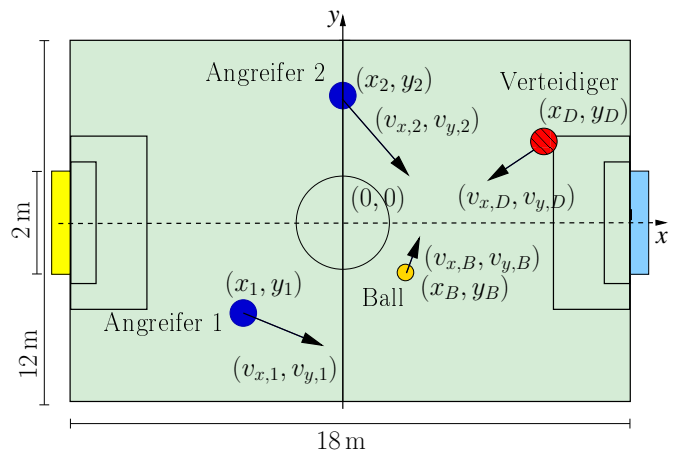


Abbildung 5.18: Kernproblem im Roboterfußball.

Bisher werden im Roboterfußball heuristische Methoden der Verhaltensprogrammierung [Ris09] zur Regelung der Kooperation eingesetzt; häufig abgeleitet von menschlichem Fußballverhalten (z.B. [DFL⁺08, Luc02]). Will man jedoch die Antwort auf Fragen nach der *optimalen* kooperativen Spielstrategie geben, kann dies nur auf Basis eines Modells geschehen, welches die spezifische Bewegungsdynamik der beteiligten Roboter hinreichend genau berücksichtigt.

Die Formulierung und Lösung eines (linearen) gemischt-ganzzahligen Optimierungsproblems gestatten es, Antworten auf verschiedenartige Fragen zu geben, wie etwa nach suboptimalen Steuersignalen oder nach „Kosten“ zum Erreichen konkreter Systemzustände.

Für die Untersuchungen wird die Problemstellung in Abb. 5.18 mit zwei angreifenden Robotern, einem Verteidiger und einem Ball betrachtet. Es wird das Problem aus Sicht der angreifenden Roboter untersucht, deren Ziel es ist, den Ball in das gegnerische Tor zu bringen.

Problembeschreibung: Betrachtet wird das Spielfeld mit den Maßen aus Abb. 5.18 und Fußballroboter der „RoboCup Middle Size League“ des RoboCup-Wettbewerbs mit omnidirektionalem Antrieb gemäß den Daten aus Tabelle 5.2.

Das dynamische Verhalten der Roboter und des Balles ändert sich vor allem dann, wenn ein Dribbling stattfindet. Eine gesicherte Ballkontrolle erfordert langsamere Geschwindigkeiten und größere Kurvenradien. Die Geschwindigkeit des frei rollenden Balles nimmt in Folge der Reibung stetig ab.

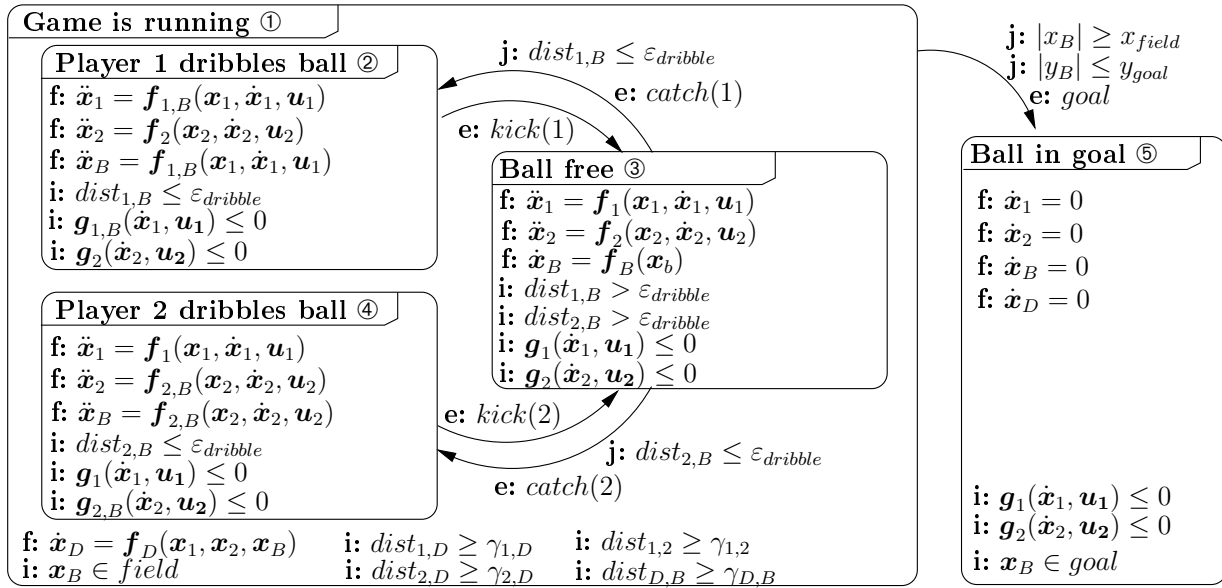


Abbildung 5.19: Hierarchisches Automatenmodell des Roboterfußball-Problems.

Untersuchte Fragestellungen: Folgende Grundfragen werden anhand des Szenarios beantwortet:

- Echtzeitfähigkeit der simultanen Aufgabenzuweisung und Bahnplanung: Was ist für einen festen Zeithorizont die beste mögliche taktische Verbesserung der Spielsituation?
- Exemplarisch zur Systemanalyse und Erreichbarkeit: Wie nahe kann der Ball dem Tor in einer bestimmten Zeit gebracht werden?

Modellierung: Die Beschreibung des Szenarios zeigt die Vorzüge der Modellierung mit hierarchischen hybriden Zustandsautomaten. Für den untersuchten Offensiv-Spielzug zeigt Abb. 5.19 dazu ein einfaches Teilmodell, das die wesentlichen physikalischen Merkmale abbildet. Die obere Hierarchieebene „Game is running“ ist genau dann aktiv, wenn sich der Ball innerhalb des Spielfeldes befindet. Sie legt allgemein gültige Beschränkungen der Kollisionsvermeidung fest und enthält ein Modell einer reaktiven Verteidigerbewegung. Auf derselben Hierarchieebene beschreibt der Zustand „Ball in goal“ den angestrebten Endzustand. Eine erweiterte Betrachtung kann beispielsweise auf dieser Ebene noch Zustände beschreiben, in denen der Ball das Feld verlässt und eine neue Positionierung der Roboter gewünscht wird.

Das Bewegungsverhalten der Roboter und des Balles wird eine Hierarchieebene tiefer in den Zuständen „Player 1 dribbles ball“, „Player 2 dribbles ball“ und „Ball free“ beschrieben. Damit werden die wesentlichen physikalischen Dynamikmodi unterschieden, wobei die Ballannahme und ein ausgeführter Schuss als Sprungbedingungen und Rückstellabbildungen beschrieben werden ($i \in \{1, 2\}$):

$$kick(i) : \dot{x}^B(t_k + 0) = 3 \cdot \dot{x}^i(t_k - 0), \quad catch(i) : x^B(t_k + 0) = x^i(t_k - 0).$$

Für alle anderen Trajektorien wird Stetigkeit beim Wechsel des Automatenzustands gefordert.

Der Ball und jeder Roboter ist durch seinen kontinuierlichen und einen diskreten Systemzustand charakterisiert, der eine bestimmte Teilaufgabe oder Rolle repräsentiert. Als spezifische Bewegungsmodi der Angreifer werden die freie Bewegung (Index 1, 2) und das Führen eines Balles (Index B) unterschieden.

Zur Beschreibung der betrachteten Roboter der „RoboCup Middle Size League“ mit omnidirektionalem Antrieb bietet sich das Punktmassenmodell an, welches mit exakter Zeitdiskretisierung in das Modell aufgenommen wird.

Für die geschaltete Balldynamik

$$\dot{\mathbf{x}}^B(t) = \mathbf{v}^B(t), \quad \mathbf{v}^B(t) = \begin{cases} c_{trac} \mathbf{v}^B(t) & , \text{ in Zustand } \textcircled{3} \\ \mathbf{v}^i(t) & , \text{ in Zustand } \textcircled{2}, \textcircled{4} \end{cases}$$

ist die Geschwindigkeit entweder durch eine reibungsabhängige Konstante $c_{trac} < 0$ oder durch die Geschwindigkeit des ballführenden Roboters $i \in \{1, 2\}$ festgelegt.

Beschränkungen an Zustands- und Steuervariablen gemäß den Bewegungsmodi werden in den Invarianten \mathbf{i} : modelliert. In diesem Beispiel sind alle Steuerungen und Geschwindigkeiten durch einfache Konstanten beschränkt

$$\mathbf{i}: g_q^i(\dot{\mathbf{x}}^i) = \|\mathbf{v}^i\|_2 - v_{q,i}^{UB} \leq 0$$

mit konstanter oberer Schranke $v_{q,i}^{UB}$ (für \mathbf{u}^i gilt entsprechendes). Ein Distanzmaß zwischen zwei Objekten i_1 und i_2 wird durch eine Hilfsvariable $dist_{i_1,i_2}$ repräsentiert und mittels einer konstanten Schranke γ_{i_1,i_2} werden Kollisionsbeschränkungen formuliert.

Für einen ballführenden Roboter wird die Beschränkung an die Maximalgeschwindigkeit um einen Faktor¹ $c_{v,dr} = 0.7$ gesenkt.

Weiterhin wird in den Sprungbedingungen die Tatsache berücksichtigt, dass ein Roboter nahe genug beim Ball sein muss, um einen Schuss auszuführen. Ein einfacher reaktiver Verteidiger, der sich stets in Richtung der aktuellen Ballposition bewegt, wird linear durch

$$\dot{\mathbf{x}}^D = \mathbf{v}_x^D(t) := \frac{v_D^{UB}}{D_{max}}(\mathbf{x}^B(t) - \mathbf{x}^D(t))$$

($\max_{x,y,t} \{|\mathbf{x}^B(t) - \mathbf{x}^D(t)|, |\mathbf{y}^B(t) - \mathbf{y}^D(t)|\} \leq D_{max} konst.; \mathbf{y}^D, v_y^D$, analog) beschrieben. v_D^{UB} ist dabei eine konstante obere Schranke für $\|\mathbf{v}^D\|_2$.

Das Zeitgitter, auf dem ein lineares Modell basiert, wird so gewählt, dass für zwei aufeinanderfolgende Takte gilt $\frac{t_{k+1}-t_k}{t_k-t_{k-1}} = 1.2$. Für jeweils 50 zufällig generierte Spielsituationen werden Vorschauhorizonte von $n_c = 5, \dots, 11$ Zeitschritten untersucht und dabei jeweils $t_f = \frac{n_c}{1.5}$ s gesetzt. Das so gewählte Zeitgitter trägt einerseits der wachsenden Unsicherheit der Prädiktion einer MPC-Regelung in dynamischen Umgebungen Rechnung und berücksichtigt andererseits, dass ein Vorschauhorizont so groß sein muss, dass kooperative Spielzüge (z.B. ein Passspiel) im Zeitraum $[t_{0|k}, t_{N|k}]$ erfolgreich ausgeführt werden können.

Die Automatenstruktur wird (wie in Abschnitt 3.1.2 vorgestellt) in Kombinationen logischer Ausdrücke mit Ungleichungen transformiert und mit Hilfe der Big-M-Technik in lineare Ungleichungen übersetzt (s. Abschnitt 2.1.1).

Distanzabhängige Beziehungen werden mit Hilfe der Gleichung (4.20) (mit $n_d = 4$ und $\varphi_0 = 0$) modelliert. Darunter fallen die Forderung, dass Ball und Roboter zur Ballannahme höchstens 0.2 m voneinander entfernt sein dürfen oder Kollisionsvermeidungsbedingungen, die wir aus Effizienzgründen nur für den Verteidiger betrachten.

¹geschätzter Wert

Um konkrete Fragestellungen zu untersuchen, können diese nun aufbauend auf die Systembeschreibung als Zielfunktion formuliert werden, deren Minimum die gesuchte Antwort charakterisiert. Dazu können alle diskreten und kontinuierlichen Zustands- und Steuervariablen des Systemmodells zu einer gewichteten Summe kombiniert werden; zum Beispiel sind dies die Werte $b_q(k)$, welche bestimmte Automatenzustände q repräsentieren, $\mathbf{x}(k)$, welche Positionen und Geschwindigkeiten enthalten und alle Hilfsvariablen, wie beispielsweise die Distanz $dist_{BD}(k)$.

a) Zur Berechnung eines optimalen taktischen Verhaltens werden folgende Komponenten betrachtet, die als gewichtete Summe zu einer Zielfunktion zusammengeführt werden:

- der diskrete Zustand „Ball in goal“

$$s_1 \sum_{k=1}^{n_c} \Delta_{t,k} b_{\textcircled{5}}(k) ,$$

- der Ballabstand zum Gegenspieler

$$s_2 \sum_{k=1}^{n_c} \Delta_{t,k} dist_{BD}(k) ,$$

- der Steueraufwand

$$s_3 \sum_{k=1}^{n_c} \Delta_{t,k} \sum_{i=1}^2 |u_x^i(k)| + |u_y^i(k)| ,$$

- der idealen Endzustand (einer der Spieler ist im Ballbesitz und hat großen Abstand zum Gegenspieler)

$$s_4(b_{\textcircled{2}}(n_c) + b_{\textcircled{4}}(n_c)) + s_5 dist_{BD}(n_c) .$$

Durch die Wahl der Gewichte s_1, \dots, s_5 lässt sich ein gewünschtes Teamverhalten bezüglich defensiver oder offensiver Auslegung parametrisieren. Dabei ist systematisch nach Prioritäten abzuwägen, so dass beispielsweise gesichert ist, dass ein erzieltes Tor ($b_{\textcircled{5}} = 1$) stets die Optimallösung darstellt. Für die in Abb. 5.20 dargestellten Ergebnisse wurde $s_1 = -42$, $s_2 = -0.2$, $s_3 = 0.001$, $s_4 = -6$ und $s_5 = -0.15$ gewählt.

b) Exemplarisch wird die Frage „Wie nahe kann der Ball dem Tor kommen, wenn jeder Angreifer wenigstens einmal den Ball spielt?“ durch Verwendung folgender Zielfunktion

$$-x^B(n_c + 1) + |y^B(n_c + 1)| + \varepsilon \cdot \sum_{k=1}^{n_c} \Delta_{t,k}^2 \sum_{i=1}^2 |u_x^i(k)| + |u_y^i(k)| , \quad (5.6)$$

unter folgenden zusätzlichen Nebenbedingungen beantwortet,

$$\sum_{k=1}^{n_c+1} b_{\textcircled{2}}(k) \geq 1 , \quad \sum_{k=1}^{n_c+1} b_{\textcircled{4}}(k) \geq 1 .$$

Der letzte Summand in (5.6) wird mit geringer Gewichtung (z.B. $\varepsilon = 0.001$) eingefügt, um sinnvolle Werte für alle Steuerungen zu erhalten. Aufgrund der linearen Struktur besitzt das Problem andernfalls unter Umständen eine große Lösungsmenge, deren Elemente die Zielfunktion gleichwertig global minimieren.

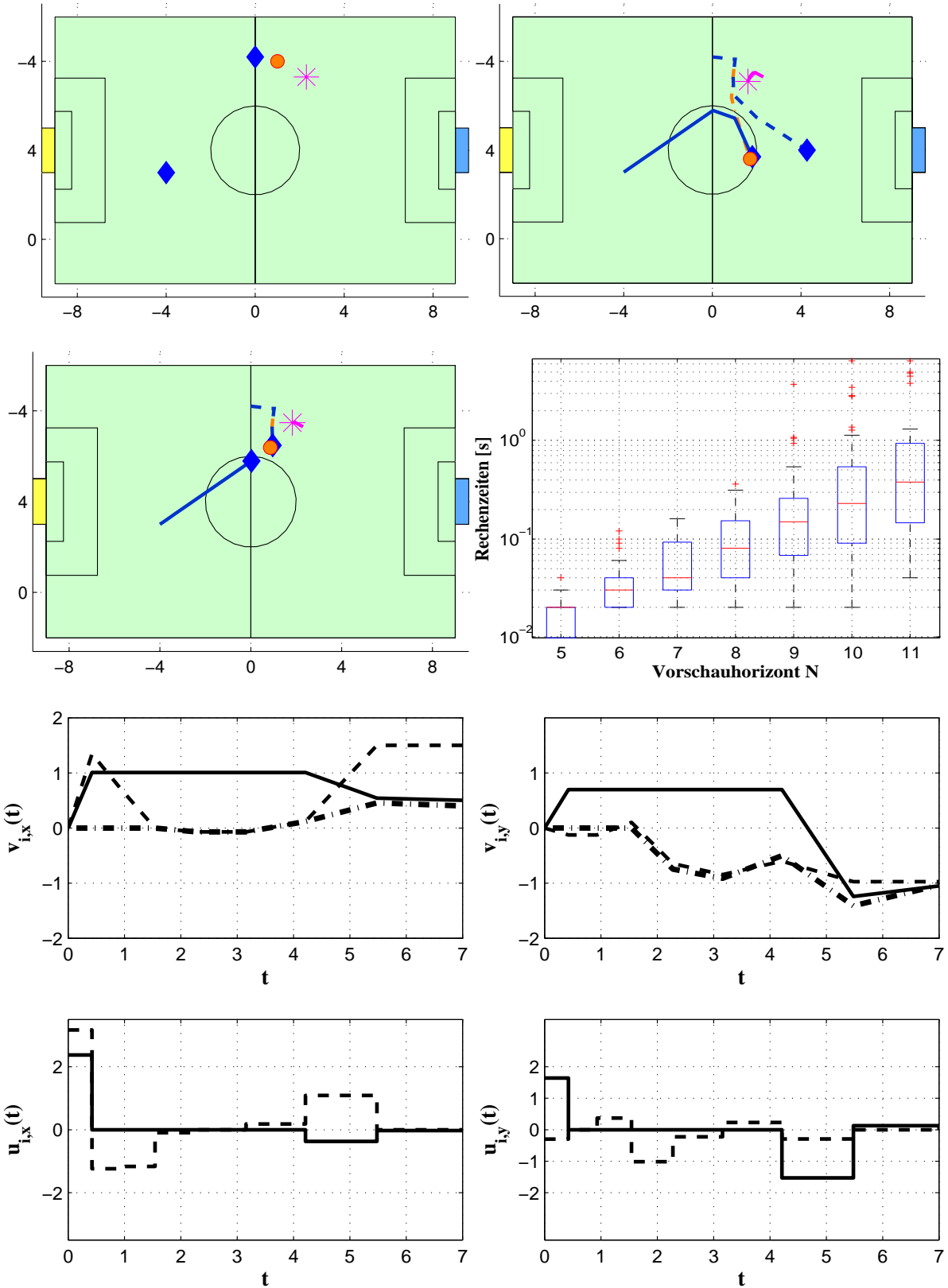


Abbildung 5.20: Berechnetes optimales kooperatives Verhalten (Zeitschritte $k = 1, 7, 9$): Der Angreifer 1 geht zum Ball, umspielt den Verteidiger und passt ihn in Richtung Mitspieler. Angreifer 2 nimmt den Ball dort an und dribbelt. Der Verteidiger (—*) folgt stets der Ballposition.

Dargestellt sind Rechenzeiten für verschiedene Vorschauhorizonte sowie optimale Trajektorien für Geschwindigkeiten und Beschleunigung der Angreifer (— und - -) und des Balles (- - -).

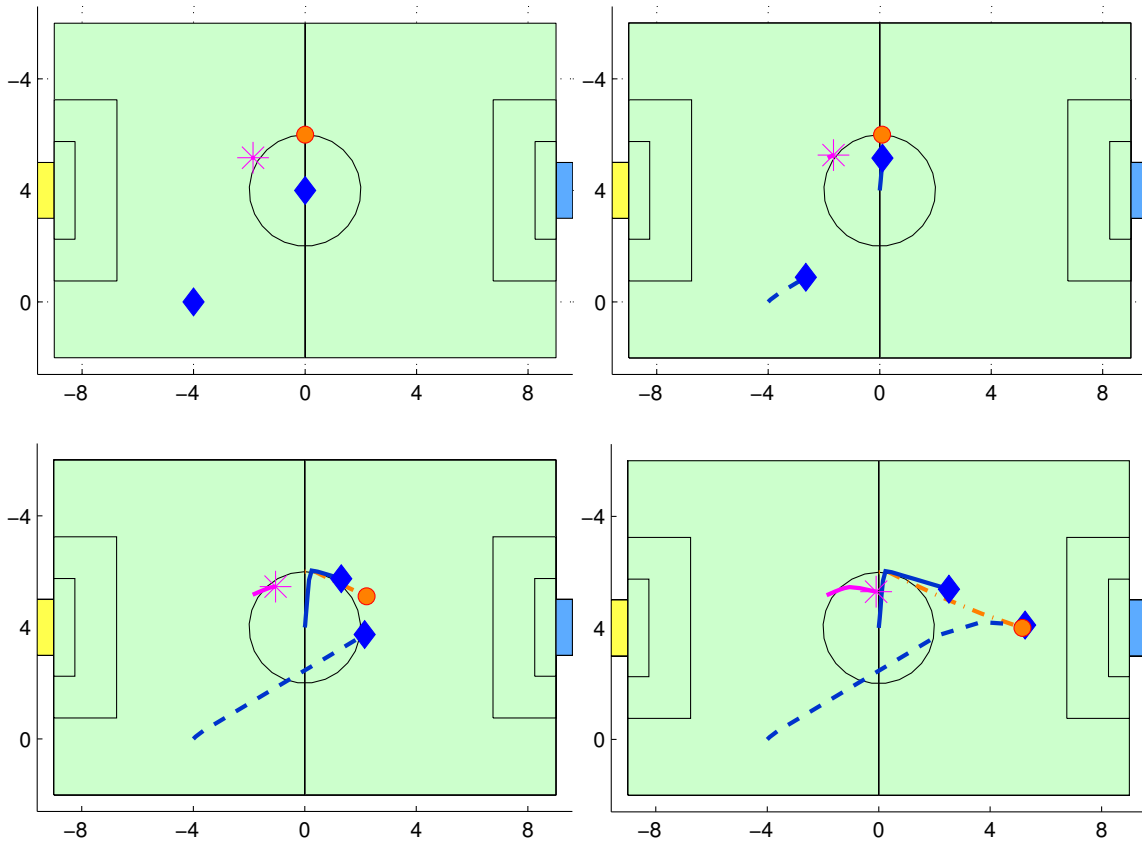


Abbildung 5.21: Optimallösung der Frage, wie nahe der Ball dem Tor innerhalb einer festen Zeitspanne gebracht werden kann. Die Darstellung für ausgewählte Zeitpunkte t_k ($k \in \{2, 6, 9, 13\}$) lässt erkennen, dass der Angreifer 1 zum Ball geht, diesen kurz dribbelt, ihn zu Mitspieler 2 passt. Der nimmt den Ball auf und schießt in die gewünschte Richtung weiter.

Ergebnisse: Die Abbildungen 5.20 zeigen exemplarisch eine Lösung zur Berechnung optimalen strategischen Verhaltens. Weiterhin ist der Rechenzeitvergleich für verschiedene Anzahlen an Zeitdiskretisierungspunkten dargestellt. Das zu lösende Minimierungsproblem für $n_v = 2$ Roboter und n_c Zeitschritte setzt sich dabei zusammen aus $(9n_v + 8)n_c + 10$ kontinuierlichen und $24n_c + 4$ binären Variablen, für die insgesamt $(42n_v + 15)n_c + 16n_v + 9$ Nebenbedingungen einzuhalten sind.

Ein exemplarisches Ergebnis zur Frage nach dem minimalen Ballabstand zum Tor ist in Abb. 5.21 und 5.22 aufgeführt. Dazu wurde das Problem für eine Zeitdauer von $t_f = 5$ s bei $n_c = 11$ Zeitschritten mit Rechenzeiten von etwa 0.4 s gelöst.

Bewertung: Die Untersuchung der Echtzeitfähigkeit optimaler taktischer Verbesserung zeigt, dass die Probleme effizient lösbar sind. Allerdings muss auch festgestellt werden, dass die Rechenzeiten offenbar in Einzelfällen deutlich über den Durchschnittswerten liegen. Eine genauere Betrachtung dieser Fälle zeigt, dass es sich dabei überwiegend um Spielsituationen handelt, in denen der Gegenspieler einen starken Einfluss auf die optimale Entscheidung nimmt. In praktischer Anwendung kann dafür entweder in entsprechenden Situationen zu einer MPC-Regelung auf kürzeren Zeithorizont umgeschaltet oder auf suboptimale Lösungen mit garantierter Rechenzeitbeschränkung zurückgegriffen werden.

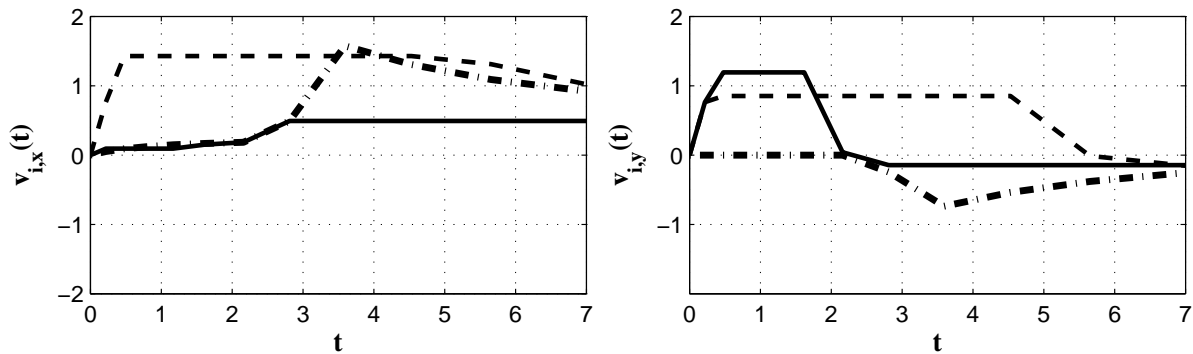


Abbildung 5.22: Geschwindigkeiten für Angreifer (— und - -) und Ball (- · -) zur berechneten Lösung aus Abbildung 5.21.

Die Einbindung der Methode in eine dezentrale Entscheidungsstruktur, in welcher der einzelne Roboter anhand eines Modells entscheidet, das nur den ihm am nächsten stehenden Mitspieler und Verteidiger berücksichtigt, ist damit grundsätzlich möglich.

Anhand eines exemplarischen Beispiels wurde weiterhin eine Anwendung zur Systemanalyse gezeigt. Allgemein können durch die optimierungsbasierte Methode alle Fragestellungen an das Modell untersucht werden, die sich durch eine Zielfunktion über kontinuierliche und binäre Variablen formulieren lassen. Dies umfasst beispielsweise auch Fragen nach der zeitlichen Entwicklung aktiver Automatenzustände. Im Gegensatz zu reinen Fragen der Erreichbarkeit erlaubt die Methode, auch Fragen nach einer besten oder schlechtesten möglichen zeitlichen Entwicklung diskreter und kontinuierlicher Zustände zu beantworten.

e

5.3.2 Kooperative Zielbeobachtung

Anhand eines allgemein anerkannten, skalierbaren, kooperativen Szenarios aus der Literatur [Par02] wird die Echtzeitfähigkeit einer MPC-Regelung auf Basis einer MLD-Beschreibung² gezeigt.

Das Problem der kooperativen Beobachtung von Zielen ist ein Grundproblem verschiedener Überwachungs- und Beobachtungs-Szenarien im Umweltmonitoring, in der Verkehrsbeobachtung und in Katastropheneinsätzen. Die Aufgabe, eine größere Menge an Sensoren oder Beobachtungspunkten mit kooperierenden Fahrzeugen in Kontakt zu halten, lässt sich nicht ohne signifikanten Optimalitätsverlust in Teilaufgaben zerlegen und zeichnet sich durch die enge Kopplung an diskreter Entscheidung und kontinuierlicher Trajektorienplanung aus.

Problembeschreibung: Betrachtet wird eine Gruppe von Fahrzeugen und eine Menge zu beobachtender, beweglicher Ziele (vgl. Abb. 5.23).

Ein Ziel gilt dann als beobachtet, sobald es sich innerhalb des Sichtbereiches eines Fahrzeugs befindet. Dieser wird als kreisförmig mit Radius R angenommen. Aufgabe der Fahrzeuge ist es stets, möglichst viele Ziele im Sichtradius zu behalten.

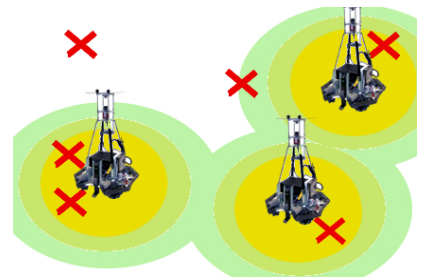


Abbildung 5.23: Kooperative Beobachtung von Zielen.

²Die Untersuchungen wurden in Zusammenarbeit mit Juliane Kuhn durchgeführt [Kuh09] und zur Veröffentlichung eingereicht [KRv11]

Untersuchte Fragestellungen: Ziel ist es auszuloten, inwieweit MILP-basierte MPC-Regelungen grundsätzlich in der Lage sind, das Kernproblem kooperativer Mobilität in Echtzeit zu lösen und somit die Voraussetzung zum Einsatz in realen Systemen zu erfüllen.

Modellierung: Zunächst wird die Problemstellung systematisch mit einem Punktmas-
senmodell zweiter Ordnung in Euler-Diskretisierung untersucht.

Um die Qualität der Lösungen im Vergleich zu den Untersuchungen aus [Par02] bewerten zu können, wird eine kreisrunde Beobachtungszone jedes Roboters mit Radius $R = 2500$ Längeneinheiten (LE) gewählt. Für alle Geschwindigkeiten gilt $v \in [-200, 500] \frac{\text{LE}}{\text{s}}$; für Beschleunigungen $u \in [-200, 500] \frac{\text{LE}}{\text{s}^2}$.

Es werden n_Z Ziele $\begin{pmatrix} \xi^j \\ \eta^j \end{pmatrix}$ ($j = 1, \dots, n_Z$) und n_v Fahrzeuge $\begin{pmatrix} x^i(t) \\ y^i(t) \end{pmatrix}$ ($i = 1, \dots, n_v$) betrachtet. Das Vorliegen einer Zielbeobachtung wird mit einer Big-M-Formulierung des Ausdrucks

$$b_{ij}(\tau) = 1 \Rightarrow \left\| \begin{pmatrix} x^i(\tau) \\ y^i(\tau) \end{pmatrix} - \begin{pmatrix} \xi^j \\ \eta^j \end{pmatrix} \right\| \leq R$$

und mit

$$\hat{b}_j(\tau) = 0 \Leftrightarrow \sum_{i=1}^{n_v} b_{ij}(\tau) \geq 1$$

modelliert, wobei je einer weitere binäre Variable $\hat{b}_j(t)$ den Zeilen zugeordnet ist. Gilt also $\hat{b}_j(t) = 0$, so ist Ziel j beobachtet.

Für die Bewertung durch eine Zielfunktion werden weiterhin wie folgt Hilfsvariablen definiert, welche den Abstand von Robotern zu unbeobachteten Zielen misst:

$$d_{ij}(\tau) := \begin{cases} \left\| \begin{pmatrix} x^i(\tau) \\ y^i(\tau) \end{pmatrix} - \begin{pmatrix} \xi^j \\ \eta^j \end{pmatrix} \right\| & , \text{ falls } \hat{b}_j(\tau) = 1 \\ 0 & , \text{ falls } \hat{b}_j(\tau) = 0 . \end{cases}$$

Der Ausdruck wird wiederum mit einem Big-M-Ansatz in eine algebraische Form gebracht. Alle Normen des Modells werden gemäß Gl. (4.20) mit $n_d = 8$ und $\varphi_0 = 0$ linearisiert.

Zur MPC-Regelung wird eine Zielfunktion konstruiert, welche die Zahl unbeobachteter Ziele, die Abstände zu unbeobachteten Zielen und den nötigen Steueraufwand minimiert. Auf einem festen Zeitgitter $t_{0|k}, t_{1|k}, \dots, t_{N|k}$ berechnen wir

$$\min s_1 \cdot \sum_{\tau=1}^N \sum_{j=1}^{n_Z} \Delta_t \hat{b}_j(\tau) + s_2 \cdot \sum_{\tau=1}^N \sum_{j=1}^{n_Z} \sum_{i=1}^{n_v} \Delta_t d_{ij}(\tau) + s_3 \cdot \sum_{\tau=1}^N \sum_{i=1}^{n_v} \Delta_t (|u_x^i(\tau)| + |u_y^i(\tau)|)$$

mit geeignet gewählten Gewichten $s_1 = 10000$, $s_2 = 0.5$ und $s_3 = 10$.

Untersucht wird der MPC-Ansatz bei sich zufällig bewegenden Zielen für verschiedene Zeithorizonte N und Gebietsgrößen.

Ergebnisse: Für eine MLD-Beschreibung der oben genannten Bedingungen wurde der MPC-Regler mit Hilfe der *Multi-Parametric Toolbox* MPT [Kva08] erstellt.

Der Regler wurde zunächst anhand einer einfachen Simulation mit festen Zielpositionen in MATLAB evaluiert, mit mehreren hundert zufälligen Positionierungen in einem Gebiet

von 8000×8000 LE. Die Qualität der Lösung wurde mit Hilfe der in [Par02] definierten „A-Metrik“

$$A = \frac{1}{n_Z} \sum_{t_s=1}^{N_T} \sum_{j=1}^{n_Z} \frac{b_j^{(t_s)}}{N_T}$$

bewertet (für eine Gesamtsimulationsdauer von N_T Zeiteinheiten, wobei die binäre Variable $b_j^{(t_s)}$ genau dann den Wert 1 besitzt, wenn Ziel j zum Zeitpunkt t_s beobachtet ist).

Für ein Verhältnis von Robotern zur Anzahl an Zielen von $n_v/n_Z = 1/2$ wurden zunächst Rechenzeiten (Abb. 5.24a) und Qualitäten (Abb. 5.24b) für wachsende Vorschauhoriizonte von MPC-Regelungen gemessen³. Dabei wurde die Lösung eines MILP über die gesamte Simulationsdauer $N_T = 20$ dem N_T -maligen Aufruf eines MPC-Reglers mit Vorschauhori-zont N gegenübergestellt.

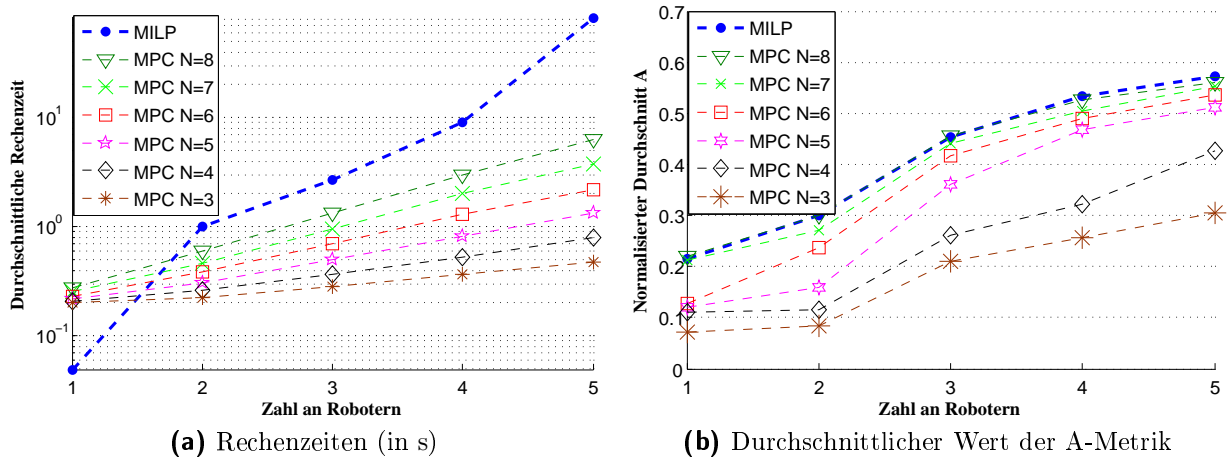


Abbildung 5.24: Test des MPC-Reglers in Abhängigkeit von Roboteranzahl und Zeithorizont.

Im Weiteren wurde anhand der Simulation ein Vergleich einer MPC-Regelung mit dem heuristischen, dezentralen Ansatz A-CMOMMT aus [Par02] gezogen. Dazu wurden zufällig positionierte Roboter und Ziele in einem zentralen Bereich (2000×2000 LE) des Operationsgebietes gewählt, so dass anfangs alle Ziele beobachtet sind. Die Zahl an Robotern zur Anzahl an Zielen standen dabei im Verhältnis $n_v/n_Z = 1/4$. Es wurden für verschiedene Ausdehnungen des Operationsgebietes (1000 LE bis 50000 LE) und Anzahlen an Robotern ($n_v = 1, \dots, 5$) jeweils 250 zufällige Situationen simuliert. Die Bewegung der Ziele erfolgte zufällig und es wurden Ergebnisse für eine Simulationsdauer von $N_T = 120$ Zeiteinheiten bewertet. Der Vorschauhori-zont betrug dabei $N = 5$ und die Beschleunigung der Roboter war auf das Intervall $[-200, 200]$ ($\frac{\text{LE}}{\text{s}^2}$) beschränkt.

Ergebnisse der Untersuchung sind in Abb. 5.25 dargestellt. Es zeigt sich, dass die Qualität der MPC-Reglung – trotz weniger nachteiliger Unterschiede in der Parametrisierung des Tests – die Güte des A-CMOMMT-Ansatzes erreicht.

Weitere Untersuchungen in [Kuh09] bestätigen die Echtzeitfähigkeit von MILP-basierten MPC-Ansätzen für dieses Problem der kooperativen Zielbeobachtung. Dies gilt ebenso für die Modellierung mit exakter Diskretisierung des Punktmassenmodells zweiter Ordnung.

³Rechner: DualCore CPU (Intel Pentium 4, 3 GHz); 3 GB RAM; MILP-Solver CPLEX.

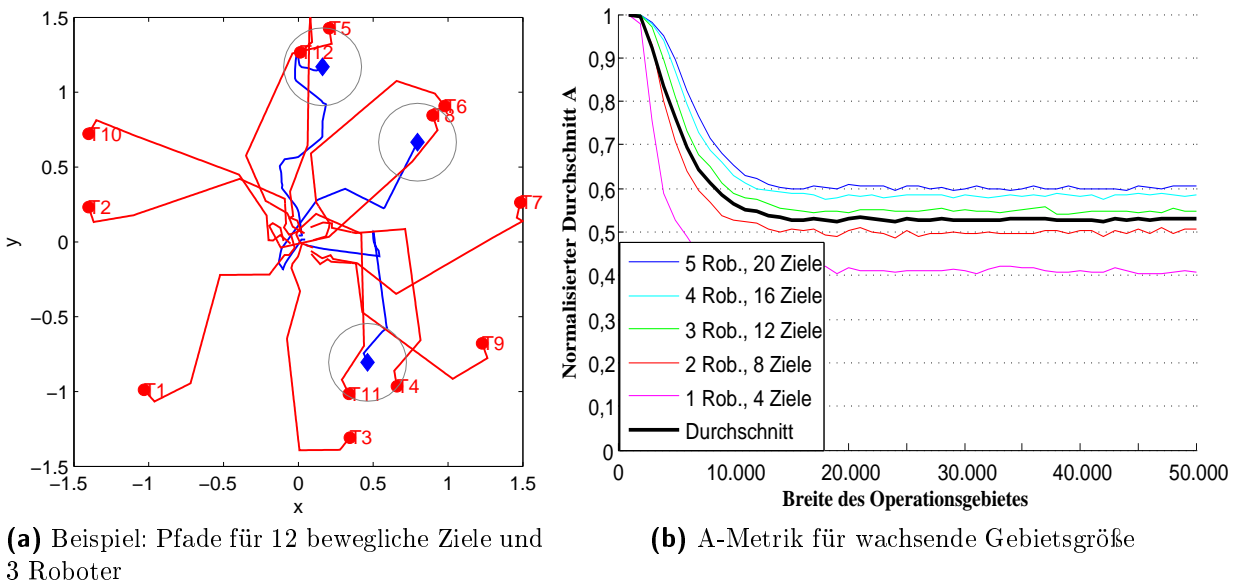


Abbildung 5.25: Ergebnisse für einen Test analog zu [Par02].

Die Berechnung eines expliziten MPC-Reglers erwies sich allerdings selbst bei übervereinfachten Beispielen für die Speicher- und Rechenkapazitäten einfacher Arbeitsplatzrechner als deutlich zu aufwendig, so dass darauf verzichtet wurde, den Ansatz für interessante Teilprobleme mit ausreichendem Vorschauhorizont weiter zu untersuchen.

Bewertung: Die Ergebnisse zeigen, dass moderate Probleme kooperativer Mobilität in Echtzeit gelöst werden können und eine Anwendung im Rahmen einer MPC-Regelung möglich ist. Entscheidend ist dabei, dass die zu lösenden Systemaufgaben mit wenigen Zeitschritten des Vorschauhorizonts hinreichend genau beschrieben werden kann.

Der Vergleich anhand eines skalierbaren Benchmarkproblems mit Lösungen aus [Par02] zeigt ferner die Güte des optimierungsbasierten Planungskonzeptes. Für zahlreiche weitere wissenschaftliche Untersuchungen der Anwendung in realen Fahrzeugsystemen, in dezentralen Entscheidungsstrukturen und unter Berücksichtigung von Unbestimmtheiten ist damit eine Erfolg versprechende Basis gelegt.

Weitere Untersuchungen zeigten, dass eine Berechnung *expliziter* MPC-Regler selbst für Minimalbeispiele zu aufwendig ist und bestätigen damit die bereits in [MB06] genannte Problematik für Systeme mit signifikantem Anteil diskreter Struktur. Weitere detaillierte Bewertungen anhand der A-Metrik finden sich in [Kuh09].

Der entwickelte Regler zeigt sich für Szenarien mit wenigen Robotern insgesamt als stabilisierend und echtzeitfähig und lässt sich darüber hinaus in eine dezentrale Reglerarchitektur einbinden.

Die *Stabilität* des modellprädiktiven Reglers kann a priori mit Hilfe von Satz 2 gesichert werden. Die dort genannte Funktion $\tilde{f}(\mathbf{x}, \mathbf{u})$ ist hier durch das MLD-System gegeben. Binäre Größen – bzw. deren Relaxationen – und kontinuierliche Größen im Steuer-Zustands-Raum sind so zu transformieren, dass der Gleichgewichtspunkt $(\mathbf{0}, \mathbf{0})$ genau die Situationen beschreibt, in denen alle Ziele beobachtet sind. Als Zielmenge \mathbb{X}_T bietet sich dann die Menge an, welche beschreibt, dass mindestens ein Ziel überdeckt ist. Die Funktionen

$F(\mathbf{x}(N+1|k))$ und $L(\mathbf{x}(\tau|k), \mathbf{u}(\tau|k))$ lassen sich dann so festlegen, dass die Beziehungen (3.8a) bis (3.8c) mit der modellprädiktiven Reglervorschrift erfüllt sind.

5.4 Aspekte der Übertragung auf reale Mehrfahrzeugsysteme

Betrachtet man reale Probleme kooperierender mobiler Fahrzeug- und Robotersysteme, die in verschiedenartigen Umgebungen mit verschiedensten Objekten interagieren, so deckt die Aufgaben- und Trajektorienplanung nur einen Teil der in der Entwicklung realer Systeme zu lösenden Probleme ab. Zahlreiche Wechselwirkungen verschiedenster physikalischer und technischer Gesetzmäßigkeiten erfordern Antworten auf ein Geflecht an Fragen, welches nur im Rahmen einer interdisziplinären Betrachtung in seiner Gesamtheit bewältigt werden kann.

Andererseits verlangt die Komplexität des Gesamtproblems aber auch danach, zunächst Antworten auf isolierte Detailfragen zu finden, die unter geeigneten Annahmen entkoppelt von übrigen Fragestellungen betrachtet werden. Somit wird als Grundlage jeder Methodenentwicklung und Entscheidungsfindung eine Basis von Annahmen abstrahiert, die sorgfältig abwägt, welche Einflüsse im betrachteten Teilproblem entscheidend und welche vernachlässigbar sind. Auf Grund der Vielschichtigkeit physikalischer und technischer Interaktion ist dies ein hoch komplexer Prozess. Darüber hinaus gibt die technische und rechnerische Machbarkeit Schranken an den Detaillierungsgrad von Basisannahmen vor.

Könnte auf Basis gewählter Annahmen eine Lösung der Teilproblemstellung gefunden werden, so bleibt auf Grund der Auswahl einer Basisannahme stets eine Unsicherheit in der Frage, ob die Lösung auch in der realen Welt Gültigkeit besitzt. Da gerade im Umfeld mobiler Robotersysteme im Außenbereich reale Testläufe meist zeitaufwändig, kostspielig, gefährlich und nur bedingt beobachtbar sind, haben sich in diesem Bereich in den vergangenen Jahren Simulationen als anerkannter effizienter Baustein des Entwicklungsprozesses etabliert. Simulationen können Experimente mit realen Systemen in realen Umgebungen nicht ganz – aber auf Grund ihrer zeitlichen Skalierbarkeit und verschleißfreien Wiederholbarkeit – zu einem großen Teil ersetzen.

Im aktuellen Forschungs- und Entwicklungsstand existieren bereits sehr detaillierte Simulationsumgebungen für Klassen von Detailproblemen, etwa zur Hardware-in-the-loop oder Software-in-the-loop Simulation von mobilen Robotersystemen [FPv08, BSCL06, CLW⁺07] und zur Simulation von Drahtlossensornetzwerken [LLWC03]. Ein breiter angelegter Überblick zu Simulationsumgebungen angrenzender wissenschaftlicher Disziplinen findet sich in [KRL⁺08].

Um die zahlreichen Wechselwirkungen realer Systeme und Umgebungen auch simulativ erfassen zu können, dürfen von einer Kopplung bestehender Simulationsumgebungen si-

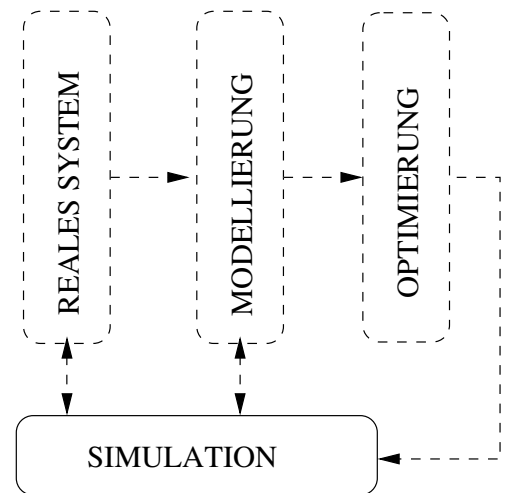


Abbildung 5.26: Simulation als begleitendes Werkzeug der Validierung im Entwicklungsprozess.

gnifikant synergetische Effekte erwartet werden. In [KRL⁺08] wurde dazu systematisch ein breit angelegtes Konzept – genannt *MM-ulator* – entwickelt, welches diese Synergieeffekte diskutiert und eine Softwarearchitektur vorstellt. Ein Erkenntnisgewinn darf immer dann erwartet werden, wenn die Simulationsumgebung die Basisannahmen der zu testenden Methode an mindestens einer Stelle in der Detailliertheit übertrifft.

Für die Beispiele der kooperativen Zielbeobachtung in Abschnitt 5.3.2 konnte somit durch die Anbindung an das *Multi-Robot-Simulation-Framework* - *MuRoSimF* [FPv08] bereits eine systematische Evaluierung und Parameteranpassung im MPC-Algorithmus durchgeführt werden [Kuh09, KRv11]. Ebenso ermöglichten systematische Testläufe in *Tossim* [LLWC03] und dessen Erweiterung *Tython* eine Evaluierung der kooperativen Datensammlung (Abschnitt 5.2.1) unter realitätsnahen Bedingungen in WSN [KSA⁺10, KRA⁺10].

Damit stellt die Simulation einen hilfreichen Begleiter in jeder Entwicklungsphase der vorgestellten Konzepte dar. Durch die Möglichkeit der Validierung und systematischen Parametrisierung von Modellen kann ein hohes Qualitätsniveau der Regleralgorithmen bereits in der Entwicklungsphase gewährleistet werden.

Zusammenfassung des Kapitels

Die Eignung approximativer, gemischt-ganzzahliger linearer Modelle zur Analyse, Planung, Systemauslegung und Echtzeitregelung kooperativer Fahrzeugsysteme wurde systematisch ausgelotet. Dabei wurden sowohl anerkannte Benchmarkprobleme als auch neue Fragestellungen untersucht und die Modelle auf Approximationsgenauigkeit und Berechnungseffizienz der anknüpfenden gemischt-ganzzahligen Optimierung geprüft.

In Anwendung auf die Benchmarkprobleme konnte dabei eine Echtzeitfähigkeit für den Einsatz in MPC-Reglerstrukturen gezeigt werden. Anhand neuer Fragestellungen wurde die breite Anwendbarkeit der Methoden sowie das Potential dargestellt, neue Antworten zu wechselwirkenden Systemanforderungen zu geben.

Exemplarisch konnte gezeigt werden, dass die approximative Methode Abschätzungen des allgemeinen nichtlinearen Problems erlaubt, die somit – zum Beispiel in Form von Startschätzungen – zur Lösung des Originalproblems herangezogen werden können.

Die Untersuchung der Berechnungseffizienz zeigte deutliche Unterschiede der Eignung einzelner Modelle und verdeutlicht mit der charakteristischen exponentiell zunehmenden Komplexität die Grenzen der Anwendung. Insbesondere wenn starke Nichtlinearitäten des Originalproblems über eine umfangreiche Hybridisierung zu einer ausgeprägten diskreten Struktur der MILP-Approximation führen, kann durch den Übergang auf das Ersatzmodell kein Effizienzgewinn mehr erzielt werden.

Die Echtzeitanwendung zur diskret-kontinuierlichen, modellprädiktiven Regelung des Kernproblems kooperativer Mobilität durch eine simultane Aufgaben- und Trajektorienplanung ist – selbst in dynamischen Umgebungen – grundsätzlich möglich.

Abschließend wurde der Übergang zu realen Mehrfahrzeugsystemen und der dabei hilfreiche Einsatz von Simulatoren in der Validierung diskutiert.

6 Zusammenfassung und Ausblick

Gegenstand der vorliegenden Arbeit ist die Modellierung, Approximation und Optimierung der Trajektorien- und Aufgabenplanung kooperierender Fahrzeuge. Die untersuchten Problemstellungen sind durch eine enge Kopplung diskreter Struktur und kontinuierlicher Dynamik charakterisiert. Im Kern der kooperativen Aufgabenzuweisung und Trajektorienplanung stehen somit schwer entscheidbare Fragestellungen.

In gängigen Planungsmethoden der praktischen Anwendung werden bislang Heuristiken für sehr spezifische Problemstellungen eingesetzt, welche die nichtlinearen bewegungsdynamischen Eigenschaften häufig nur in sehr grober Approximation berücksichtigen. Eine Verallgemeinerung bestehender Ansätze bezüglich Fahrzeugklassen und Aufgabenstellungen ist deshalb meist überhaupt nicht oder nur sehr eingeschränkt möglich.

Ausgehend von allgemeinen Betrachtungen der Theorie hybrider dynamischer Systeme und der diskret-kontinuierlichen Optimalsteuerung (Kapitel 2 und 3) wird in der vorliegenden Arbeit ein umfassendes Modellierungskonzept auf Basis hierarchischer hybrider Automaten entwickelt. Über ein Approximationskonzept mit linearen Ersatzmodellen und über geeignete Transformationen der logischen Struktur wird die Planung optimaler Trajektorien und Aufgaben kooperierender Fahrzeuge effizienten Verfahren der gemischt-ganzzahligen, linearen Optimierung zugänglich gemacht.

Dieses Konzept erlaubt es, die charakteristische, diskret-kontinuierliche Kopplung der Probleme in der Planung zu berücksichtigen und damit eine – in diesem Umfang neue – simultane Aufgaben- und Trajektorienplanung durchzuführen. Wegen der mit der Anzahl diskreter Variablen exponentiell wachsenden Berechnungskomplexität gemischt-ganzzahliger Optimierungsprobleme ist eine im konkreten Anwendungsfall spezifische Abwägung von Genauigkeit in den Teilmodellen und angestrebter Effizienz nötig. Durch verfahrensspezifische Parameter, wie etwa in der Wahl des Zeitgitters, der Anzahl von Linearisierungspunkten oder des Detaillierungsgrades von Approximationspolygonen, lassen sich die vorgestellten Ersatzmodelle und Approximationen skalieren. Dies erlaubt eine aufgaben- und fahrzeugdynamikspezifische Anpassung und ermöglicht damit einen einheitlichen Zugang zur Untersuchung unterschiedlicher Fragestellungen kooperativer Trajektorien- und Aufgabenplanung.

Da das vorgeschlagene Approximationskonzept auf einer Beschreibung mit logisch verknüpften, linearen Ausdrücken als Nebenbedingungen eines gemischt-ganzzahligen Minimierungsproblems beruht, ist die Erweiterung um verschiedene approximative Teilmodelle (s. Kapitel 4) unproblematisch. Dies ist vor allem in den numerischen Vorzügen der linearen Optimierung begründet, speziell in der Eigenschaft, weitestgehend unabhängig von Startwerten, effizient globale Optima zu berechnen.

Für verschiedene Aspekte neuer Anwendungen von Mehrfahrzeugkooperationen werden bestehende Modellierungsansätze integriert und erweitert (z. B. wechselnde Dynamikmodi und Hindernisvermeidung) sowie angepasste neue Modelle zur Beschreibung unterlagerter Formationsstrukturen, von Umgebungen und physikalischen Anforderungen der Drahtloskommunikation entwickelt (Kapitel 4).

Die Formulierung von Zielfunktionen auf Basis sowohl diskreter als auch kontinuierlicher Steuer- und Zustandsgrößen gestattet eine flexible Berücksichtigung diskreter Ereignisse einerseits und von Verläufen kontinuierlicher Fahrzeugtrajektorien andererseits. Die Lösung der betrachteten diskret-kontinuierlichen Optimierungsprobleme liefert für viele Fragestellungen kooperativer Mehrfahrzeugsysteme effizient berechenbare Näherungslösungen und kann in der vorliegenden Form bereits in Spezialfällen zu einer neuen, echtzeitfähigen modellprädiktiven Regelung der kooperativen Aufgabenzuweisung und Trajektorienplanung herangezogen werden.

Aus Sicht der Anwendung wird eine neue, leistungsfähige Methode vorgestellt, die es erlaubt, heterogene Fahrzeugflotten mit wechselnden Fahrdynamikmodi in dynamischen Umgebungen zu berücksichtigen und eine optimierungsbasierte Trajektorien- und Aufgabenplanung durchzuführen, welche zielorientierte, koordinierte Fahrzeugbewegungen unter garantierten, wechselnden Konnektivitätseigenschaften ermöglicht.

In Kapitel 5 wird die Leistungsfähigkeit der entwickelten Modelle und Methoden zur effizienten Planung demonstriert. Dabei wird sowohl die Qualität einzelner Modelle als auch die Eignung der gemischt-ganzzahligen linearen Optimierung zur simultanen, diskret-kontinuierlichen Entscheidungsfindung in kooperativen Fahrzeugsystemen systematisch untersucht. Dazu werden unterschiedliche Benchmarkprobleme der Wegpunktzuweisung, Zielbeobachtung und des Roboterfußballs betrachtet und Lösungen zu neuartigen Fragestellungen wie der kooperativen Mobilität verschiedener kommunizierender Fahrzeuge in strukturierten Gebieten oder zur Unterstützung in Drahtlossensornetzwerken präsentiert.

Die vorgestellten Konzepte ermöglichen die Berechnung wichtiger Abschätzungen, die dem Entwickler in der Systemauslegung und in der heuristischen Verhaltensprogrammierung aussagekräftige Referenzlösungen zur Verfügung stellen. Unter ausreichender Berücksichtigung der physikalischen Fahrzeugdynamik und der charakteristischen diskret-kontinuierlichen Kopplung von Systemzuständen können quantitative Erreichbarkeitsfragen über den zeitlichen Verlauf sowie Worst-case-Fragestellungen beantwortet werden.

Eine vielversprechende Fortführung stellt die Weiterentwicklung der in der Arbeit vorgestellten Methoden für dezentrale oder verteilte Reglerstrukturen unter Einbeziehung von Modellunsicherheiten und Synchronisation dar. Ferner erscheint ein Ausbau zu einer in sich konsistenten Mehr-Hierarchien-Lösung mit weiteren Approximationsschichten interessant, welche Näherungsmodelle mit einem wachsenden Grad an Nichtlinearitäten – insbesondere in den fahrdynamischen Eigenschaften – ermöglichen könnten.

Anhang

A Zentrale Sätze der Optimalsteuerung

Formulierung nach [Glo05]:

Satz 3. (*Pontryaginsches Maximumprinzip*)

Sei $(\mathbf{x}^*, \mathbf{u}^*, q_0^*, \dots, q_{n_c}^*)$ ein Prozess, der das Optimalsteuerungsproblem (3.1) bis (3.2g) minimiert. Sei

$$\begin{aligned}\tilde{H}(\mathbf{x}, \mathbf{u}, q) &:= L_{q_k}(\mathbf{x}(t), \mathbf{u}(t)) + \boldsymbol{\lambda}(t)^T \mathbf{f}_{q_k}(\mathbf{x}(t), \mathbf{u}(t)) + \\ &\quad + \boldsymbol{\mu}_h(t)^T \mathbf{g}_{q_k}(\mathbf{x}(t), \mathbf{u}(t)) + \boldsymbol{\mu}_g(t)^T \mathbf{h}_{q_k}(\mathbf{x}(t), \mathbf{u}(t))\end{aligned}$$

die erweiterte Hamiltonfunktion, dann existieren Multiplikatorfunktionen $\boldsymbol{\lambda} : [t_0, t_f] \rightarrow \mathbb{R}^{n_\lambda}$, $(\boldsymbol{\mu}_h, \boldsymbol{\mu}_g) =: \boldsymbol{\mu} : [t_0, t_f] \rightarrow \mathbb{R}^{n_\mu}$ und Multiplikatoren $\boldsymbol{\nu} \in \mathbb{R}^{n_\nu}$, so dass folgende Bedingungen aus der Variationsrechnung gelten:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f} \\ \dot{\boldsymbol{\lambda}} &= -\tilde{H}_{\mathbf{x}} \quad (\text{wobei: } \tilde{H}_{\mathbf{x}} := \frac{\partial \tilde{H}}{\partial \mathbf{x}}) \\ \tilde{H}_{\mathbf{u}} &= 0 \\ \tilde{H}_q &= 0 \\ 0 &= \mu_{h,i} h_k \\ 0 &\leq \boldsymbol{\mu}_h \\ 0 &= (\Phi_{0,\mathbf{x}(t_0)} + \mathbf{r}_{0,\mathbf{x}(t_0)}^T \boldsymbol{\nu}_0 + \boldsymbol{\lambda}(t_0)) \delta \mathbf{x}_0 \\ 0 &= (\Phi_{0,(t_0)} + \mathbf{r}_{0,(t_0)}^T \boldsymbol{\nu}_0 - \tilde{H} \Big|_{t_0}) \delta t_0 \\ 0 &= (\Phi_{n_c,\mathbf{x}(t_f)} + \mathbf{r}_{n_c,\mathbf{x}(t_f)}^T \boldsymbol{\nu}_{n_c} + \boldsymbol{\lambda}(t_f)) \delta \mathbf{x}_f \\ 0 &= (\Phi_{n_c,(t_f)} + \mathbf{r}_{n_c,(t_f)}^T \boldsymbol{\nu}_{n_c} - \tilde{H} \Big|_{t_f}) \delta t_f \\ \boldsymbol{\lambda}(t_k^-) &= \boldsymbol{\lambda}(t_k^+) \quad (\text{wobei: } t_k \text{ Zeitpunkt eines Phasenübergangs}) \\ \tilde{H} \Big|_{t_k^-} &= \tilde{H} \Big|_{t_k^+} \\ \boldsymbol{\lambda}(t_k^-) &= \Phi_{n_c,\mathbf{x}(t_k)} + \mathbf{r}_{k,\mathbf{x}(t_k)}^T \boldsymbol{\nu}_i + \boldsymbol{\lambda}(t_k^+) \\ \tilde{H} \Big|_{t_k^-} &= \Phi_{n_c,(t_k)} + \mathbf{r}_{k,t_k}^T \boldsymbol{\nu}_k + \tilde{H} \Big|_{t_k^+} .\end{aligned}$$

Weiterhin gilt für fast alle $t \in [t_0, t_f]$

$$L_{q_k^*}(\mathbf{x}^*, \mathbf{u}^*) + \boldsymbol{\lambda}^T \mathbf{f}_{q^*}(\mathbf{x}^*, \mathbf{u}^*) = \min_{\mathbf{u}(t)} L_{q_k}(\mathbf{x}^*, \mathbf{u}, q) + \boldsymbol{\lambda}^T \mathbf{f}_q(\mathbf{x}^*, \mathbf{u}) .$$

Satz 4. (*Bellmansches Optimalitätsprinzip / Hamilton-Jacobi-Bellman-Gleichung*)

Eine optimale Entscheidungsfolge hat die Eigenschaft, dass für einen beliebigen Anfangszustand $\mathbf{x}(0)$ und beliebige (eventuell nicht-optimale) Entscheidungen $\mathbf{u}(0)$, die übrigen Entscheidungsfunktionen eine optimale Entscheidungsfolge bilden bezüglich des Zustandes $\mathbf{x}(1)$, der sich aus der Entscheidung $\mathbf{u}(0)$ ergibt.

Für den kontinuierlichen Fall wird die Wertefunktion $V : \mathbb{X} \times [t_0, t_f] \rightarrow \mathbb{R}$ definiert als

$$V(\mathbf{x}, t) = \min_{\mathbf{x}, \mathbf{u}} \left\{ \int_t^{t_f} L(\mathbf{x}(\theta), \mathbf{u}(\theta)) d\theta \right\},$$

welche die Mindestkosten zu Überführung eines Zustandes $\mathbf{x}(t)$ in einen Zustand $\mathbf{x}(t_f)$ beschreibt.

Der fundamentale Satz der dynamischen Programmierung bedient sich der Wertefunktion und besagt die Gültigkeit der sogenannten Hamilton-Jacobi-Bellman-Gleichung

$$-V_t = \min_{\mathbf{x}, \mathbf{u}} \{ L(\mathbf{x}(t), \mathbf{u}(t)) + V_{\mathbf{x}}^T \mathbf{f}_q(\mathbf{x}(t), \mathbf{u}(t)) \}.$$

B Definitionen zur Systemstabilität (nach [Laz06])

Definition 10. (Funktionen der Klasse \mathcal{K})

Eine reellwertige skalare Funktion $\alpha : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ gehört zur Klasse \mathcal{K} ($\alpha \in \mathcal{K}$) falls sie stetig und monoton wachsend ist und $\alpha(0) = 0$ gilt.

Definition 11. (λ -Kontraktivität und positiv invariante Menge)

Sei $0 \leq \lambda \leq 1$. Eine Menge $\mathcal{P} \subseteq \mathbb{R}^{n_x}$ die den Ursprung im Inneren enthält heißt λ -kontraktive Menge des Systems $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k))$, falls $\forall \mathbf{x} \in \mathcal{P}$ gilt, dass $\mathbf{f}(\mathbf{x}) \in \lambda \cdot \mathcal{P}$. Für $\lambda = 1$ heißt eine λ -kontraktive Menge *positiv invariant*.

Definition 12. (Attraktivität und Stabilität nach Lyapunov) Sei $\mathbf{x}^* \in \mathbb{R}^{n_x}$ ein Gleichgewichtspunkt des Systems $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k))$ (d.h. $\mathbf{x}^* = \mathbf{f}(\mathbf{x}^*)$) und sei $\mathbb{X} \subseteq \mathbb{R}^{n_x}$ eine Menge, welche eine offene Nachbarschaft von \mathbf{x}^* enthält.

1. Der Gleichgewichtspunkt \mathbf{x}^* ist *Lyapunov-stabil*, falls $\forall \varepsilon > 0 \exists \delta(\varepsilon) > 0$ so dass

$$\|\mathbf{x}(0) - \mathbf{x}^*\| \leq \delta \Rightarrow \|\mathbf{x}(k) - \mathbf{x}^*\| \leq \varepsilon \text{ für alle } k \in \mathbb{N},$$

wobei $\mathbf{x}(k)$ der Systemzustand des Systems $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k))$ zur Zeit $k \in \mathbb{N}$ mit Startzustand $\mathbf{x}(0)$ ist.

2. Der Gleichgewichtspunkt \mathbf{x}^* heißt *attraktiv in \mathbb{X}* , falls

$$\lim_{k \rightarrow \infty} \|\mathbf{x}(k) - \mathbf{x}^*\| = 0 \text{ für alle } \mathbf{x}(0) \in \mathbb{X}$$

3. Der Gleichgewichtspunkt \mathbf{x}^* ist *lokal attraktiv*, falls $\exists \delta > 0$, so dass

$$\|\mathbf{x}(0) - \mathbf{x}^*\| \leq \delta \Rightarrow \lim_{k \rightarrow \infty} \|\mathbf{x}(k) - \mathbf{x}^*\| = 0$$

4. Der Gleichgewichtspunkt \mathbf{x}^* ist *global attraktiv*, falls er in \mathbb{R}^{n_x} attraktiv ist.
5. Der Gleichgewichtspunkt \mathbf{x}^* ist *im Lyapunov-Sinn asymptotisch stabil in \mathbb{X}* , falls er sowohl Lyapunov-stabil und attraktiv in \mathbb{X} ist.
6. Der Gleichgewichtspunkt \mathbf{x}^* ist *im Lyapunov-Sinn lokal (global) asymptotisch stabil*, falls er sowohl lokal (global) asymptotisch stabil im Lyapunov-Sinn und lokal (global) attraktiv ist.
7. Der Gleichgewichtspunkt \mathbf{x}^* ist *exponentiell stabil in \mathbb{X}* , falls $\exists \theta > 0, \tau \in [0, 1)$, so dass

$$\|\mathbf{x}(k) - \mathbf{x}^*\| \leq \theta \|\mathbf{x}(0) - \mathbf{x}^*\| \tau^k \text{ für alle } \mathbf{x}(0) \in \mathbb{X} \text{ und für alle } k \in \mathbb{N}$$

8. Der Gleichgewichtspunkt \mathbf{x}^* ist *lokal exponentiell stabil*, falls $\exists \delta > 0, \theta > 0$ und $\tau \in [0, 1)$, so dass

$$\|\mathbf{x}(0) - \mathbf{x}^*\| \leq \delta \Rightarrow \|\mathbf{x}(k) - \mathbf{x}^*\| \leq \theta \|\mathbf{x}(0) - \mathbf{x}^*\| \tau^k, \text{ für alle } k \in \mathbb{N}$$

9. Der Gleichgewichtspunkt \mathbf{x}^* ist *global exponentiell stabil*, falls er exponentiell stabil in \mathbb{R}^{n_x} ist

Literaturverzeichnis

- [ADE⁺01] ALUR, R., T. DANG, J. M. ESPOSITO, R. B. FIERRO, Y. HUR, F. IVANCIC, V. KUMAR, I. LEE, P. MISHRA, G. J. PAPPAS und. O. SOKOLSKY: *Hierarchical Hybrid Modeling of Embedded Systems*. In: *EMSOFT*, Seiten 14–31, 2001.
- [ADG07] ASARIN, E., T. DANG und. A. GIRARD: *Hybridization methods for the analysis of nonlinear systems*. *Acta Informatica*, 43(7):451–476, February 2007.
- [AHL00] ALUR, R., T. A. HENZINGER, G. LAFFERRIERE und. G. J. PAPPAS: *Discrete abstractions of hybrid systems*. *Proceedings of the IEEE*, 88(7):971–984, July 2000.
- [AIS06] AZUMA, S., J. IMURA und. T. SUGIE: *Lebesgue Piecewise Affine Approximation of Nonlinear Systems and Its Application to Hybrid System Modeling of Biosystems*. In: *Proc. 45th IEEE Conference on Decision and Control*, Seiten 2128–2133, 13–15 Dec. 2006.
- [AMP] AMPL OPTIMIZATION LLC: *AMPL - A Modeling Language for Mathematical Programming*. <http://www.ampl.com>.
- [ASB07] ALTHOFF, M., O. STURSBURG und. M. BUSS: *Safety Assessment of Autonomous Cars using Verification Techniques*. In: *Proc. American Control Conference ACC '07*, Seiten 4154–4159, 9–13 July 2007.
- [Bal85] BALAS, E.: *Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems*. *SIAM Journal on Algebraic and Discrete Methods*, 6(3):466–486, 1985.
- [BBM00] BEMPORAD, A., F. BORRELLI und. M. MORARI: *Optimal controllers for hybrid systems: Stability and piecewise linear explicit form*. In: *Proceedings of the 39th IEEE Conference on Decision and Control*, Band 2, Seiten 1810 – 1815, 2000.
- [BCM09] BULLO, F., J. CORTÉS und. S. MARTÍNEZ: *Distributed Control of Robotic Networks*. *Applied Mathematics Series*. Princeton University Press, 2009. To appear. Electronically available at <http://coordinationbook.info>.
- [BF06] BRANCA, C. und. R. FIERRO: *A Hierarchical Optimization Algorithm for Cooperative Vehicle Networks*. In: *American Control Conference, June 14-16, 2006, Minneapolis, Minnesota, USA*, 2006.
- [BFTM00] BEMPORAD, A., G. FERRARI-TRECATE und. M. MORARI: *Observability and controllability of piecewise affine and hybrid systems*. *Automatic Control, IEEE Transactions on*, 45(10):1864–1876, Oct. 2000.
- [BFV06] BORRELLI, F., P. FALCONE und. C. D. VECCHIO: *Event-Based Receding Horizon Control for Two-Stages Multi-Product Production Plants*. In: *American Control Conference, June 14-16, 2006, Minneapolis, Minnesota, USA*, 2006.

- [BG03] BEMPORAD, A. und. N. GIORGETTI: *Logic-Based Hybrid Solvers for Optimal Control of Hybrid Systems*. In: *42th IEEE Conf. on Decision and Control, Maui, Hawaii, USA*, Seiten 640–645, 2003.
- [BG06] BEMPORAD, A. und. N. GIORGETTI: *Logic-based methods for optimal control of hybrid systems*. IEEE Trans. Automatic Control, 51:963–976, 2006.
- [BGH⁺02] BUSS, M., M. GLOCKER, M. HARDT, O. VON STRYK, R. BULIRSCH und. G. SCHMIDT: *Nonlinear hybrid dynamical systems: modeling, optimal control, and applications*. In: S. ENGELL, G. F. E. S. (Herausgeber): *Modelling, Analysis and Design of Hybrid Systems*, Band 279 der Reihe *Lecture Notes in Control and Information Sciences*, Seiten 311–335, Berlin, Heidelberg, 2002. Springer-Verlag.
- [BGLM05] BISWAS, P., P. GRIEDER, J. LÖFBERG und. M. MORARI: *A Survey on Stability Analysis of Discrete-Time Piecewise Affine Systems*. In: *IFAC World Congress*, Prague, Czech Republic, Juli 2005.
- [BIP05] BELTA, C., V. ISLER und. G. PAPPAS: *Discrete Abstractions for Robot Motion Planning and Control in Polygonal Environments*. IEEE Transactions on Robotics, 21:864–874, 2005.
- [BJSS07] BORGES DE SOUSA, J., K. H. JOHANSSON, J. SILVA und. A. SPERANZON: *A verified hierarchical control architecture for coordinated multi-vehicle operations*. International Journal of Adaptive Control and Signal Processing, 21(2–3):159–188, 2007. Special issue on autonomous adaptive control of vehicles.
- [BM99] BEMPORAD, A. und. M. MORARI: *Control of systems integrating logic, dynamics, and constraints*. Automatica, 35(3):407–427, 1999.
- [Bor03] BORELLI, F.: *Constrained Optimal Control of Linear and Hybrid Systems*, Band 290 der Reihe *Lecture Notes in Control and Information Sciences*. Springer Berlin / Heidelberg, 2003.
- [BR04] BASU, P. und. J. REDI: *Movement control algorithms for realization of fault-tolerant ad hoc robot networks*. IEEE Network, 18(4):36–44, 2004.
- [BSCL06] BALAKIRSKY, S., C. SCRAPPER, S. CARPIN und. M. LEWIS: *USARSim: providing a framework for multi-robot performance evaluation*. In: *Proceedings of PerMIS*, 2006.
- [Chr06] CHRISTOPHERSEN, F. J.: *Optimal Control and Analysis for Constrained Piecewise Affine Systems*. Dr. sc. ETH Zurich thesis, ETH Zurich, Zurich, Switzerland, August 2006.
- [CKC04] CHAIMOWICZ, L., V. KUMAR und. M. CAMPOS: *A Paradigm for Dynamic Coordination of Multiple Robots*. Autonomous Robots, 17(1):7–21, 2004.
- [CLW⁺07] CARPIN, S., M. LEWIS, J. WANG, S. BALAKIRSKY und. C. SCRAPPER: *USARSim: a robot simulator for research and education*. In: *Proc. of the 2007 IEEE Intl. Conf. on Robotics and Automation*, 2007.
- [CPPSV06] CARLONI, L., R. PASSERONE, A. PINTO und. A. SANGIOVANNI-VINCENTELLI: *Languages and Tools for Hybrid Systems Design*. Foundations and Trends in Design Automation, 1(1):1–204, 2006.

- [Cv04] COLLINS, P. und. J. H. VAN SCHUPPEN: *Observability of Piecewise-Affine Hybrid Systems*. In: *Hybrid Systems: Computation and Control*, Band 2993 der Reihe *Lecture Notes in Computer Science*, Seiten 265–279. Springer Berlin / Heidelberg, 2004.
- [DFL⁺08] DYLLA, F., A. FERREIN, G. LAKEMEYER, J. MURRAY, O. OBST, T. RÖFER, S. SCHIFFER, F. STOLZENBURG, U. VISSER und. T. WAGNER: *Approaching a formal soccer theory from behaviour specifications in robotic soccer*. In: DABNICHCKI, P. und. A. BACA (Herausgeber): *Computers in Sport*, Kapitel 6, Seiten 161–185. WIT Press, London, 2008.
- [DM06] DAS, T. und. R. MUKHERJEE: *An Extension of the Minimum Principle with Application to Switched Linear Systems*. In: *American Control Conference, June 14-16, 2006, Minneapolis, Minnesota, USA*, 2006.
- [DS01] DIAS, M. und. A. STENTZ: *A Market Approach to Multirobot Coordination*. Technischer Bericht, Robotics Institute, Carnegie Mellon University, August 2001.
- [DS09a] DOLD, J. und. O. STURSBURG: *A Communication-Based Distributed Model Predictive Control Scheme with Time-Varying Consistent Constraints*. In: *The European Control Conference*, 2009.
- [DS09b] DOLD, J. und. O. STURSBURG: *Distributed predictive control of communicating and platooning vehicles*. In: *Proc. 48th IEEE Conference on held jointly with the 2009 28th Chinese Control Conference Decision and Control CDC/CCC 2009*, Seiten 561–566, 2009.
- [Ear05] EARL, M. G.: *Optimization techniques for multi-vehicle cooperative control*. Doktorarbeit, Cornell University, 2005.
- [Eck08] ECKE, C.: *Sensorik, Regelung und Realisierung eines unbemannten schwimmenden Fahrzeuges*. Diplomarbeit, Technische Universität Darmstadt, Department of Computer Science, 2008.
- [ED04] EARL, M. G. und. R. D’ANDREA: *Iterative MILP Methods for Vehicle Control Problems*. In: *IEEE Conference on Decision and Control*, Band 4, Seiten 4369–4374, December 2004.
- [ED07] EARL, M. G. und. R. D’ANDREA: *Multi-Vehicle Cooperative Control Using Mixed Integer Linear Programming*. In: SHAMMA, J. S. (Herausgeber): *Cooperative Control of Distributed Multi-Agent Systems*, Seiten 231–259. John Wiley & Sons, Ltd, Dec 2007.
- [EFS06] ELBASSIONI, K., A. V. FISHKIN und. R. SITTERS: *On Approximating the TSP with Intersecting Neighborhoods*. In: *Algorithms and Computation*, Band 4288 der Reihe *Lecture Notes in Computer Science*, Seiten 213–222. Springer Berlin / Heidelberg, 2006.
- [FBA06] FINDEISEN, R., L. BIEGLER und. F. ALLGÖWER (Herausgeber): *Assessment and Future Directions of Nonlinear Model Predictive Control*. Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin, 2006.
- [FBDWD04] FURUKAWA, T., F. BOURGAULT, H. F. DURRANT-WHYTE und. G. DISSANAYAKE: *Dynamic allocation and control of coordinated UAVs to engage multiple targets in a time-optimal manner*. In: *IEEE Intl. Conf. on Robotics & Automation*, Band 3, Seiten 2353–2358, New Orleans, 2004.

- [Fie73] FIEDLER, M.: *Algebraic connectivity of graphs*. Czech. Math. J., 23:298–305, 1973.
- [FMSS07] FURBACH, U., J. MURRAY, F. SCHMIDSBERGER und. F. STOLZENBURG: *Hybrid Multiagent Systems with Timed Synchronization – Specification and Model Checking*. In: *Proceedings of the 5th Workshop on Programming Multi-Agent Systems (ProMAS07)*, 2007. Held in conjunction with AAMAS 2007.
- [FPv08] FRIEDMANN, M., K. PETERSEN und. O. VON STRYK: *Simulation of multi-robot teams with flexible level of detail*. In: CARPIN, S., I. NODA, E. PAGELLO, M. REGGIANI und. O. VON STRYK (Herausgeber): *Simulation, Modeling and Programming for Autonomous Robots (SIMPAN 2008)*, Nummer 5325 in *Lecture Notes in Artificial Intelligence*, Seiten 29–40, Venice, Italy, November 2008. Springer.
- [FTEBJ06] FERRARI-TRECCATE, G., M. EGERSTEDT, A. BUFFA und. M. JI: *Laplacian Sheep: A Hybrid, Stop-Go Policy for Leader-Based Containment Control*. *Hybrid Systems: Computation and Control*. Hybrid Systems: Computation and Control, Seiten 212–226, 2006.
- [FTMLM01] FERRARI-TRECCATE, G., M. MUSELLI, D. LIBERATI und. M. MORARI: *Identification of piecewise affine and hybrid systems*. In: *American Control Conference, 2001. Proceedings of the 2001*, Band 5, Seiten 3521–3526 vol.5, 2001.
- [Ger05] GERDTS, M.: *Solving mixed-integer optimal control problems by branch&bound: a case study from automobile test-driving with gear shift*. *Optimal Control Applications and Methods*, 26:1–18, 2005.
- [GHAL05] GANCET, J., G. HATTENBERGER, R. ALAMI und. S. LACROIX: *Task Planning and Control for a multi-UAV system: architecture and algorithms*. In: *IEEE International Conference on Intelligent Robots and Systems, Edmonton (Canada)*, 2005.
- [GJP08] GIRARD, A., A. A. JULIUS und. G. J. PAPPAS: *Approximate simulation relations for hybrid systems*. *Discrete Event Dynamic Systems*, 18(2):163–179, Jun 2008.
- [GL03] GROSSMANN, I. und. S. LEE: *Generalized Convex Disjunctive Programming: Non-linear Convex Hull Relaxation*. *Computational Optimization and Applications*, 26(1):83–100, 2003.
- [Glo05] GLOCKER, M.: *Diskret-kontinuierliche Optimalsteuerung: Modellierung, Numerik und Anwendung bei Mehrfahrzeugsystemen (Discrete-Continuous Optimal Control: Models, Numerical Methods and Applications in Multi-Vehicle Systems)*. Fortschritt-Berichte VDI, Technische Universität Darmstadt, October 21 2005.
- [GM02] GERKEY, B. und. M. MATARIĆ: *Sold!: Auction methods for multirobot coordination*. *IEEE Trans. on Robotics & Automation*, 18(5):758–768, 2002.
- [GMS02] GILL, P., W. MURRAY und. M. SAUNDERS: *SNOPT: An SQP algorithm for large-scale constrained optimization*. *SIAM J. Optim.*, 12:979–1006, 2002.
- [GO09] GUROBI OPTIMIZATION, I.: *Gurobi Optimizer Reference Manual, Version 2.0*. <http://www.gurobi.com>, 2009.
- [GP09] GRÜNE, L. und. J. PANNEK: *Practical NMPC suboptimality estimates along trajectories*. *Systems & Control Letters*, 58:161–168, 2009.

- [GRvS06] GLOCKER, M., C. REINL und. O. VON STRYK: *Optimal task allocation and dynamic trajectory planning for multi-vehicle systems using nonlinear hybrid optimal control*. In: *Proc. 1st IFAC-Symposium on Multivehicle Systems*, Seiten 38–43, Salvador, Brazil, October 2-3 2006.
- [GT02] GROSSGLAUSER, M. und. D. N. C. TSE: *Mobility increases the capacity of ad hoc wireless networks*. IEEE/ACM Trans. Netw., 10(4):477–486, Aug. 2002.
- [GvS02] GLOCKER, M. und. O. VON STRYK: *Hybrid optimal control of motorized traveling salesmen and beyond*. Seiten 987–992, Barcelona, Spain, July 21-26 2002. Elsevier Science.
- [HDB01] HEEMELS, W., B. DE SCHUTTER und. A. BEMPORAD: *Equivalence of hybrid dynamical models*. Automatica, 37(7):1085–1091, Juli 2001.
- [Hen96] HENZINGER, T.: *The Theory of Hybrid Automata*. In: *Proceedings of the 11th Annual Symposium on Logic in Computer Science*, Seiten 278–292, New Brunswick, NJ, 1996. IEEE Computer Society Press.
- [HHWT95] HENZINGER, T. A., P.-H. HO und. H. WONG-TOI: *HyTech: The Next Generation*. In: *IEEE Real-Time Systems Symposium*, Seiten 56–65, 1995.
- [Hod07] HODRUS, T. E.: *Prozessführungsstrategien für hybride Systeme*. Doktorarbeit, Universität Karlsruhe, 2007.
- [Hoo07] HOOKER, J. N.: *Integrated Methods for Optimization*, Band 100 der Reihe *International Series in Operations Research & Management Science*. Springer US, 2007.
- [HW04] HICKEY, T. J. und. D. K. WITTENBERG: *Using Analytic CLP to Model and Analyze Hybrid Systems*. In: *FLAIRS Conference*. AAAI Press, 2004.
- [IBM] IBM ILOG: *OPL Development Studio* (<http://www.ilog.com>).
- [ILO07] ILOG: *ILOG CPLEX 11.0, User's Manual*, 2007.
- [JV88] JONKER, R. und. T. VOLGENANT: *An Improved Transformation of the Symmetric Multiple Travelling Salesman Problem*. Operations Research, 36(1):163–167, 1988.
- [Kal02] KALLRATH, J.: *Gemischt-ganzzahlige Optimierung: Modelle und Anwendungen*. Vieweg, 1 2002.
- [Käm06] KÄMPER, J.: *Numerical Computation of Lower Bounds for Optimal Control by Semi-Infinite Programming*. Diplomarbeit, Technische Universität Darmstadt, 2006.
- [KBF⁺08] KEVICZKY, T., F. BORRELLI, K. FREGENE, D. GODBOLE und. G. BALAS: *Decentralized Receding Horizon Control and Coordination of Autonomous Vehicle Formations*. Band 16, Seiten 19–33, Jan. 2008.
- [KFB⁺06] KEVICZKY, T., K. FREGENE, F. BORRELLI, G. J. BALAS und. D. GODBOLE: *Coordinated Autonomous Vehicle Formations: Decentralization, Control Synthesis and Optimization*. In: *American Control Conference, June 14-16, 2006, Minneapolis, Minnesota, USA*, 2006.
- [KRA⁺10] KHELIL, A., C. REINL, B. AYARI, F. K. SHAIKH, P. SZCZYTOWSKI, A. ALI und. N. SURI: *Sensor Cooperation for a Sustainable Quality of Information*. In: *Pervasive Computing and Networking*. Wiley & Sons (– Zur Veröffentlichung angenommen –), 2010.

- [KRL⁺08] KROPFF, M., C. REINL, K. LISTMANN, K. PETERSEN, K. RADKHAH, F. K. SHAIKH, A. HERZOG, A. STROBEL, D. JACOBI und. O. VON STRYK: *MM-ulator: Towards a Common Evaluation Platform for Mixed Mode Environments*. In: CARPIN, S., I. NODA, E. PAGELLO, M. REGGIANI und. O. VON STRYK (Herausgeber): *Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN)*, Nummer 5325 in *Lecture Notes in Artificial Intelligence*, Seiten 41–52. Springer, November 2008.
- [KRSH06] KUWATA, Y., A. RICHARDS, T. SCHOUWENAARS und. J. P. HOW: *Decentralized Robust Receding Horizon Control for Multi-Vehicle Guidance*. In: *American Control Conference, June 14-16, 2006, Minneapolis, Minnesota, USA*, 2006.
- [KRv11] KUHN, J., C. REINL und. O. VON STRYK: *Predictive Control for Multi-Robot Observation of Multiple Moving Targets Based on Discrete-Continuous Linear Models*. In: *(eingereicht: 18th IFAC World Congress)*, 2011.
- [Krz07] KRZYSZTOF R. APT, M. W.: *Constraint Logic Programming Using Eclipse*. Cambridge University Press, Cambridge, UK, 2007.
- [KSA⁺10] KHELIL, A., F. K. SHAIKH, A. ALI, N. SURI und. C. REINL: *Delay-Tolerant Monitoring of Mobility-Assisted Wireless Sensor Networks*. In: *Delay Tolerant Networks: Protocols and Applications*. Auerbach Publications, CRC Press, Taylor & Francis Group, 2010.
- [Kuh09] KUHN, J.: *Model-Predictive Control of Cooperative Multi-Vehicle Systems Based on Discrete-Time Linear Systems*. Diplomarbeit, Technische Universität Darmstadt, 2009.
- [Kva08] KVASNICA, M.: *Efficient software tools for control and analysis of hybrid systems*. Doktorarbeit, Eidgenössische Technische Hochschule ETH Zürich, 2008.
- [Lan99] LANDSTORFER, F. M.: *Wave Propagation Models for the Planning of Mobile Communication Networks*. 1999.
- [Laz06] LAZAR, M.: *Model Predictive Control of Hybrid Systems: Stability and Robustness*. Doktorarbeit, Technische Universiteit Eindhoven, 2006.
- [LBR97] LABINAZ, G., M. M. BAYOUMI und. K. RUDIE: *A Survey Of Modeling And Control Of Hybrid Systems*. *Annual Reviews of Control*, 21:79–92, 1997.
- [LH04] LI, N. und. J. C. HOU: *Topology control in heterogeneous wireless networks: problems and solutions*. In: *Proc. IEEE INFOCOM*, 2004.
- [LHWP06] LAZAR, M., W. HEEMELS, S. WEILAND und. A. BEMPORAD: *Stabilizing Model Predictive Control of Hybrid Systems*. 51(11):1813–1818, Nov. 2006.
- [LJ06] LI, D. und. J. JOSE B. CRUZ: *Better Cooperative Control with Limited Look-Ahead*. In: *American Control Conference, June 14-16, 2006, Minneapolis, Minnesota, USA*, 2006.
- [LLWC03] LEVIS, P., N. LEE, M. WELSH und. D. CULLER: *TOSSIM: accurate and scalable simulation of entire TinyOS applications*. In: *Proc. of the 1st Intl. Conf. on Embedded Networked Sensor Systems*, Seiten 126–137, 2003.

- [LPS99] LYGEROS, J., G. J. PAPPAS und. S. SASTRY: *An introduction to hybrid system modeling, analysis, and control*. First Nonlinear Control Network Pedagogical School, Athens, Greece, pp. 307-329, 1999.
- [Luc02] LUCCHESI, M.: *Coaching the 3-4-1-2 and 4-2-3-1*. Reedswain Publishing, 2002.
- [Lun06] LUNZE, J.: *Regelungstechnik 2: Mehrgrößensysteme, Digitale Regelung*. Springer, March 2006.
- [Mag05] MAGATAO, L.: *Mixed Integer Linear Programming and Constraint Logic Programming towards a Unified Modeling Framework*. Doktorarbeit, The Federal Center of Technological Education of Parana, May 2005.
- [Mat] MATHWORKS, T.: *Matlab 7.3*.
- [MB03] MORARI, M. und. M. B. B. F. BORRELLI: *Hybrid Systems Modeling and Control*. European Journal of Control, 9:177–189, 2003.
- [MB06] MORARI, M. und. M. BARIC: *Recent developments in the control of constrained hybrid systems*. Computers & Chemical Engineering, 30(10–12):1619–1631, September 2006.
- [MFK06] MEHNE, H., M. H. FARAHİ und. A. V. KAMYAD: *MILP modelling for the time optimal control problem in the case of multiple targets*. Optimal Control Applications and Methods, 27(2):77–91, September 2006.
- [Mit07] MITCHELL, J. S. B.: *A PTAS for TSP with neighborhoods among fat regions in the plane*. In: *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, Seiten 11–18, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.
- [MS98] MARRIOTT, K. und. P. J. STUCKEY: *Programming with Constraints*. MIT Press, Cambridge, MA, London, 1998.
- [Mur07] MURRAY, R. M.: *Recent Research in Cooperative Control of Multi-Vehicle Systems*. Journal of Dynamics, Systems, Measurement and Control, 129:571–583, 2007.
- [MZKP08] MICHAEL, N., M. M. ZAVLANOS, V. KUMAR und. G. J. PAPPAS: *Maintaining Connectivity in Mobile Robot Networks*. In: *iser*, Athens, Greece, jul 2008.
- [NB07] NOTARSTEFANO, G. und. F. BULLO: *Network abstract linear programming with application to minimum-time formation control*. In: *Proc. 46th IEEE Conference on Decision and Control*, Seiten 927–932, 12–14 Dec. 2007.
- [NDH04] NEGENBORN, R., B. DE SCHUTTER und. J. HELLENDORF: *Multi-agent model predictive control: A survey*. Technischer Bericht 04-010, Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, Dezember 2004.
- [NSBJ06] NOTARSTEFANO, G., K. SALVA, F. BULLO und. A. JADBABAIE: *Maintaining limited-range connectivity among second-order agents*. In: *American Control Conference, June 14-16, 2006, Minneapolis, Minnesota, USA*, 2006.
- [Old05] OLDENBURG, J.: *Logic-based modelling and optimization of discrete-continuous dynamic systems*. Doktorarbeit, RWTH Aachen, 2005.

- [Par98] PARKER, L.: *ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation*. IEEE Trans. on Robotics & Automation, 14(2):220–240, 1998.
- [Par02] PARKER, L. E.: *Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets*. Auton. Robots, 12(3):231–255, 2002.
- [Par08] PARKER, L.: *Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems*. Journal of Physical Agents, 2(2):5–14, 2008.
- [Pet62] PETRI, C. A.: *Kommunikation mit Automaten*. Doktorarbeit, Bonn: Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962. Second Edition; New York: Griffiss Air Force Base, Technical Report RADC-TR-65–377, Vol.1, 1966, Pages: Suppl. 1, English translation.
- [PW06] PODELSKI, A. und. S. WAGNER: *Model Checking of Hybrid Systems: From Reachability Towards Stability*. In: *Hybrid Systems: Computation and Control, 9th International Workshop, Santa Barbara, CA, USA, 2006, Proceedings*, Band 3927 der Reihe *Lecture Notes in Computer Science*, Seiten 507–521. Springer, 2006.
- [RD05] ROSS, M. und. C. N. D’SOUZA: *A Hybrid Optimal Control Framework for Mission Planning*. Journal of Guidance, Control and Dynamics, vol.28 no.4:686–697, 2005.
- [RGvS09] REINL, C., M. GLOCKER und. O. VON STRYK: *Optimalsteuerung kooperierender Mehrfahrzeugsysteme*. at-Automatisierungstechnik, 57(6):296–305, 2009.
- [RH02] RICHARDS, A. und. J. P. HOW: *Aircraft Trajectory Planning With Collision Avoidance Using Mixed Integer Linear Programming*. In: *ACC*, Seiten 1936–1941. IEEE, May 2002.
- [Ris09] RISLER, M.: *Behavior Control for Single and Multiple Autonomous Agents Based on Hierarchical Finite State Machines*. Doktorarbeit, Technische Universität Darmstadt, May 15 2009.
- [RRSv08] REINL, C., F. RUH, F. STOLZENBURG und. O. VON STRYK: *Multi-Robot Systems Optimization and Analysis Using MILP and CLP*. In: *AAMAS08-Workshop on Formal Models and Methods for Multi-Robot Systems*, Estoril, Portugal, May 12-16 2008.
- [Rub86] RUBIO, J. E.: *Control and optimization; the linear treatment of non-linear problems*. manchester University Press, Oxford Road, Manchester M13 9PL, UK, 1986.
- [Sag05] SAGER, S.: *Numerical methods for mixed-integer optimal control problems*. Doktorarbeit, Universität Heidelberg, 2005.
- [SBC06] SPINELLI, W., P. BOLZERN und. P. COLANERI: *A Note on Optimal Control of Autonomous Switched Systems on a Finite Time Interval*. In: *American Control Conference, June 14-16, 2006, Minneapolis, Minnesota, USA*, 2006.
- [SBF05] SAVLA, K., F. BULLO und. E. FRAZZOLI: *On traveling salesperson problems for Dubins’ vehicle: stochastic and dynamic environments*. In: *IEEE Conf. on Decision and Control and European Control Conference*, Seiten 4530–4535, Seville, Spain, Dezember 2005.
- [SBF09] SAVLA, K., F. BULLO und. E. FRAZZOLI: *Traveling Salesperson Problems for a Double Integrator*. Automatic Control, IEEE Transactions on, 54(4):788–793, April 2009.

- [SC07] SHAIKH, M. S. und. P. E. CAINES: *On the Hybrid Optimal Control Problem: Theory and Algorithms*. Automatic Control, IEEE Transactions on, 52(9):1587–1603, 2007.
- [SE02] STURSBERG, O. und. S. ENGELL: *Optimal Control of Switched Continuous Systems Using Mixed-Integer Programming*. In: *15th IFAC World Congress, Barcelona*, 2002.
- [SFB08] SAVLA, K., E. FRAZZOLI und. F. BULLO: *Traveling Salesperson Problems for the Dubins vehicle*. IEEE Transactions on Automatic Control, 53(6):1378–1391, July 2008.
- [SFH06] SCHOUWENAARS, T., E. FERON und. J. HOW: *Multi-Vehicle Path Planning for Non-Line of Sight Communication*. In: *American Control Conference*, June 2006.
- [SG08] SUGIHARA, R. und. R. K. GUPTA: *Improving the Data Delivery Latency in Sensor Networks with Controlled Mobility*. In: *Distributed Computing in Sensor Systems*, Band 5067 der Reihe *Lecture Notes in Computer Science*, Seiten 386–399. Springer Berlin / Heidelberg, 2008.
- [SM04] SPANOS, D. und. R. MURRAY: *Robust connectivity of networked vehicles*. In: *43rd IEEE Conference on Decision and Control, 2004. CDC*, Band 3, Seiten 2893–2898, 2004.
- [SMY⁺] SUZUKA, A., R. MIYASHIRO, A. YOSHISE, T. MATSUI, A. AVIDOR und. U. ZWICK: *Dash Associates, XPRESS-MP User Guide and Reference Manual*, Dash Associates,. <http://www.dashoptimization.com>.
- [Sol87] SOLOMON, M. M.: *Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints*. Operations Research, 35 (2):254–265, 1987.
- [SRKC00] SILVA, B. I., K. RICHESON, B. H. KROGH und. A. CHUTINAN: *Modeling and verification of hybrid dynamical system using CheckMate*. In: *Proceedings of 4th International Conference on Automation of Mixed Processes*,, Seiten 323–328, 2000.
- [SRS05] SHIMA, T., S. J. RASMUSSEN und. A. G. SPARKS: *UAV Cooperative Multiple Task Assignments using Genetic Algorithms*. American Control Conference, June 2005.
- [SSKE01] SILVA, B. I., O. STURSBERG, B. H. KROGH und. S. ENGELL: *An assessment of the current status of algorithmic approaches to the verification of hybrid systems*. In: *Proc. 40th IEEE Conference on Decision and Control*, Band 3, Seiten 2867–2874, 4–7 Dec. 2001.
- [Stu00] STURSBERG, O.: *Analyse gesteuerter verfahrenstechnischer Prozesse durch Diskretisierung*. Doktorarbeit, Universität Dortmund, 2000.
- [Sus99] SUSSMANN, H. J.: *A maximum principle for hybrid optimal control problems*. Band 1, Seiten 425–430, Dec 1999.
- [TB] TORRISI, F. und. M. BAOTIC: *CPLEXINT - Matlab interface for the CPLEX solver*.
- [TB04] TORRISI, F. und. A. BEMPORAD: *HYSDEL - A tool for generating computational hybrid models*. IEEE Trans. Contr. Systems Technology, 12(2):235–249, March 2004.
- [TILT09] TEKDas, O., V. ISLER, J. LIM und. A. TERZIS: *Using Mobile Robots to Harvest Data from Sensor Fields*. In: *Proc. IEEE Wireless Communications*, 2009.

- [TOBD06] THOMAS, J., S. OLARU, J. BUISSON und. D. DUMUR: *Robust model predictive control for piecewise affine systems subject to bounded disturbances*. In: CASSANDRAS, C., A. GIUA, C. SEATZU und. J. ZAYTOON (Herausgeber): *ADHS06: 2nd IFAC Conference on Analysis and Design of Hybrid Systems*, Seiten 329–334, Alghero (Italy), 2006. Elsevier.
- [vSG00] STRYK, O. VON und. M. GLOCKER: *Decomposition of mixed-integer optimal control problems using branch and bound and sparse direct collocation*. In: S. ENGELL, S. KOWALEWSKI, J. Z. (Herausgeber): *ADPM 2000 - The 4th International Conference on Automation of Mixed Processes: Hybrid Dynamic Systems*, Seiten 99–104, Aachen, September 18-19 2000. Shaker.
- [vSG01] STRYK, O. VON und. M. GLOCKER: *Numerical mixed-integer optimal control and motorized traveling salesmen problems*. APII-JESA (Journal européen des systèmes automatisés - European Journal of Control), 35(4):519–533, 2001.
- [WB96] WILLIAMS, H. P. und. S. C. BRAILSFORD: *Advances in Linear and Integer Programming*, Kapitel Computational logic and integer programming, Seiten 249–281. Oxford University Press, Inc., New York, NY, USA, 1996.
- [WB06] WÄCHTER, A. und. L. T. BIEGLER: *On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming*. Mathematical Programming, 106(1):25–57, 2006.
- [WHW07] WANG, Y., H. DANG und. H. WU: *A survey on analytic studies of delay-tolerant mobile sensor networks*. Journal of Wireless Communications and Mobile Computing (WCMC) Special Issue on Disruption Tolerant Networking for Mobile or Sensor Networks, 7:1197–1208, 2007.
- [YOS07] YUAN, B., M. ORLOWSKA und. S. SADIQ: *On the Optimal Robot Routing Problem in Wireless Sensor Networks*. IEEE Transactions on Knowledge and Data Engineering, 19(9):1252–1261, Sept. 2007.
- [ZKS03] ZELINSKI, S., T. J. KOO und. S. SASTRY: *Hybrid System Design for Formations of Autonomous Vehicles*. In: *Proceedings. 42nd IEEE Conference on Decision and Control*, Band 1, Seiten 1–6, 2003.
- [ZP05] ZAVLANOS, M. M. und. G. J. PAPPAS: *Controlling Connectivity of Dynamic Graphs*. In: *44th IEEE Conference on Decision and Control, Seville, Spain, December 2005*, Seiten 6388–6393, 2005.

Lebenslauf

Name	Christian Karl Reinl
Anschrift	Bahnhofplatz 8 64839 Münster bei Dieburg
Geburtsdatum	07. 11. 1978
Geburtsort	Weiden in der Oberpfalz
Staatsangehörigkeit	deutsch
09.1990-06.1999	Gymnasium Neustadt an der Waldnaab Erlangung der allgemeinen Hochschulreife
10.2000-11.2005	Studium der Technomathematik an der Universität Bayreuth Abschluss als Diplom-Mathematiker (Dipl.-Math.)
12.2005-06.2007	Wissenschaftlicher Mitarbeiter am Fachgebiet Simulation, Systemoptimierung und Robotik des Fachbereichs Informatik an der Technischen Universität Darmstadt
07.2007-06.2010	Promotionsstudent im Graduiertenkolleg „Cooperative, Adaptive and Responsive Monitoring in Mixed Mode Environments“ eingerrichtet an der Technischen Universität Darmstadt